1985

A simulation for the evaluation of tolerances in a sequential manufacturing operation /

Jill B. Wootten
Lehigh University

A Simulation for the Evaluation of Tolerances in a Sequential Manufacturing Operation

by

Jill B. Wootten

A Thesis

Presented to the Graduate Committee

of Lehigh University

in Candidacy for the Degree of

Master of Science

in

Industrial Engineering

Lehigh University

1985

This thesis is accepted and approved in partial fulfillment of the requirements for the degree of Master of Science.

5/16/85
(date)

Professor in Charge

Chairman of Department

Acknowledgements

I would like to take this opportunity to thank Dr. Louis J. Plebani for all the guidance and personal instruction he has given me throughout my graduate studies. I would also like to thank my mother, father and Alan for being patient and understanding and for giving me the confidence to pursue my goals.

Table of Contents

List of Figures

1. Abstract

A user configurable simulation is presented which models sequential manufacturing operations with imbedded inspections stations. The machines are defined by sets of modifying equations which are used to alter specific attributes of a part as it is processed. When a part leaves a machine, an inspection may be performed to determine if the attibutes are within specified tolerances. If one of the attributes is out of tolerance, it is recorded as a reject and is removed from the line if so instructed. The number of rejections and statistics on an attribute are collected for each attribute modified by each machine. Data for a histogram for each attribute may also be collected. Output reports are generated from the statistics and histogram data that is compiled and can be constructed to fit the specific needs of the user. These reports provide the basis for critical decisions such as changing a distribution or tolerance range. The simulation can be easily modified to accommodate these decisions. The user, in effect, has the ability to create and modify his own simulation which is easy to implement and is informative as well.

2. Problem Definition

This model is designed to simulate a production system consisting of a series of machines which alter the configuration of the part being manufactured. Each machine is defined by a set of distributions or modifying equations which are used to alter specific attributes of a part as it is processed. Every time a part leaves a particular machine an inspection may be performed to determine if all the attributes modified by the preceding process are within specified tolerances. If any one attribute is not within its particular tolerance, the part is recorded as a reject and if so instructed, removed from the line (see Figure 2A). The number of rejections and statistics on an attribute, such as the mean and standard deviation, are collected for each attribute for each machine. A histogram of the range of values of an attribute may also be collected. These provide the basis for output reports.

The main objective of this project is to provide the user with a means of creating his own simulation of a specific manufacturing operation that is easy to understand, implement and modify as well as being accurate and informative.

Any information which is entered by the user may be modified at a later date, including: the processes and their descriptions, the attributes and their descriptions,

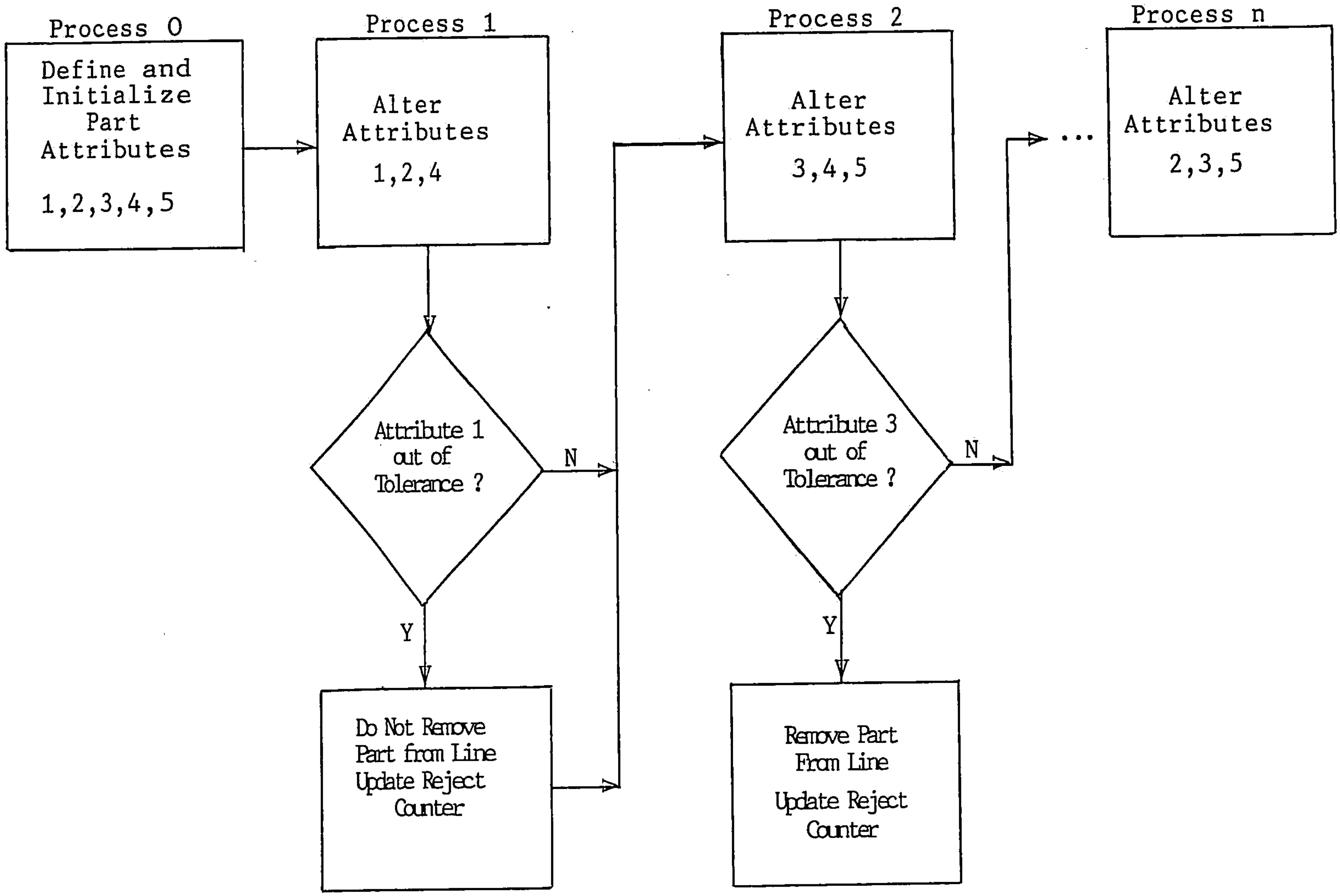

Figure 2A - Schematic Diagram of the Processes modifying equations, tolerances values, histogram specifications and the number of parts to be processed. A process and the attributes it modifies may be deleted or inserted anywhere in the system. Modifying equations, tolerances and histogram specifications may be added, deleted or modified for each attribute.

The user has the option of including inspection stations after each process is completed; these may also be modified at a later date. The user also has the option of rejecting parts out of tolerance or allowing them to continue in the system.

While the simulation is running, the user is provided with progress reports on the number of parts already processed and the total number of rejects. Statistics are collected for each attribute for each process and histogram data is collected for the particular attributes specified by the user. These provide the basis for critical decisions such as changing a distribution or tolerance range. The simulation can be easily modified to conform to these decisions.

Output reports can be constructed to accommodate the specific needs of the user. Summary reports, which contain statistical information about the processes and the attributes, and histogram tables may be generated. A report may contain all the data collected or only a specific part.

Each report may be printed on the printer and/or displayed on the screen.

In summary, this model creates a simulation based on user input. It provides information on how well the machines are performing and allows the user to adjust and redefine the processes appropriately.

3. Simulation Design

The principal intention when designing the model was to make it as interactive and user modifiable as possible. This is accomplished by using an input file to control the simulation. The file contains all the necessary information such as the processes and the attributes they modify, distributions, tolerance ranges, number of parts to be processed and histogram specifications. This data is entered interactively and then used to feed the simulation. Once the simulation is stopped, statistical information about the attributes for each process is stored in a user specified output file. This file can then be used to generate various reports on how well the machines are operating. This design provides a versatile model which can be used to simulate virtually any type of production system that consists of a series of manufacturing machines. The only stipulation is that the modifying equations or distributions must be able to be repesented as FORTRAN statements. A pictorial representation of this design is presented below and a description of each block follows.

```
 ______________         ______________
!              !       !              !
! interactive  !       !    input     !
!              !------>!              !
!   session    !       !     file     !
!______________!       !______________!
                         !    !   .. !
                         !    !      !
                         V    V      V
                       _________________
                      !                 !
                      !   simulation    !
                      !_________________!
                               !
                               !
                               V
                         ______________         ______________
                        !              !       !              !
                        !    output    !       ! statistical  !
                        !              !------>!              !
                        !     file     !       !   reports    !
                        !______________!       !______________!
```

3.1 <u>Interactive Session</u>

Once the system is booted and the date and time have been entered, the user enters the file name "RUNINPUT". This is a batch file which executes two programs. One of these programs, "PUTINFIL", enables the user to enter and/or modify an input file and the second, "CREATPRO", uses the input file to construct subroutine PROCESS, a routine which represents the manufacturing operation. This subroutine is then complied and linked to the simulation program. The same procedure is followed when an old input file is to be used without modification.

A few seconds after the name "PUTINFIL" has been entered, a title and reminder on special details will appear on the screen. Among these details is an explanation of the use of the "ESC" key and the use of a carriage return for long lists. Pressing the "ESC" key enables the user to temporarily or perminently interrupt the flow of the program. Once the routine is activated the user is presented with four options:

1) to Start Over
2) to get HELP (a list of instructions)
3) to Quit the Program
5) to Continue from where "ESC" was initiated

An "ESC" is accepted any time the user is requested to make a choice between a number of options while entering or modifying the input file. A ctrl-C can always be used to exit the program, however any data entered during the session will be lost.

A carriage return is needed to continue the program whenever a list or message appears on the screen and to continue displaying a list which is too long to be shown all at once. This is done to insure that the user has time to read all lists and messages carefully.

As soon as the special reminders have been read, the name of the input file to be used must be determined. The following options are presented:

1) Load an OLD Input File
2) Create a NEW Input File
3) List all old Input Files
Enter (1,2 or 3):

If an old file name is entered, the list of old files is searched to determine if the name actually exists. If is does not, an error message is displayed and the user is asked for another name. If the name of a new input file is entered and the name already exists as an old file, an error message is displayed. The user has the option of using the file name regardless of the fact that it already exists. Another message, which warns of the old file's destruction, is displayed if the user chooses to use the existing file name. If the new name does not exist, several checks are made to be certain that the file name adheres to the guidelines presented: the name can be up to eight alphanumeric characters, the first character must be a letter and no special characters may be used. If an error exists, the approriate message appears and the user is asked to enter another name.

Once the file name has been determined, either a new file is constructed or an old file is modified. If a new file is to be constructed, the user is given a series of prompts to enter information regarding the part attributes and their initial values, the processes and the attributes they modify, histogram specifications for particualar attributes and the number of parts to be processed. Modifying equations, tolerance values and additional FORTRAN statements, which further describe the system, may also be entered. As the information is being typed in, several opportunities are given to modify the data so that the chance of an error remaining when subroutine PROCESS is compiled, is greatly reduced. An example of the dialogue which typically occurs is presented in Figure 3.1A.

If an old file is to be modified, the following options appear on the screen:

1) Modify a Process
2) Add a Process
3) Delete a Process
4) Change the number of Parts to be Processed
5) Finished Modifying

Enter (1,2,3,4 or5):

Enter the Part Attributes

Once the new file name has been determined, the following is displayed:

```
Enter the Description for Each Attribute
of the Part being Manufactured

Attribute #   Description
     1        LENGTH
     2        HIEGHT
     3        WIDTH
     4        DONE
```

You are given the attribute # as a prompt and are expected to enter the description of each one. When you are finished entering all the part attributes, type "DONE" for the description.

You must then enter the initial values for these attributes. The screen appears as follows:

```
Enter the Initial Value (Equation A(#)=value)
of the Part being Manufactured
                                                Initial Value
Attribute #   Description                       A(1)=.03571
     1        LENGTH                            A(2)=.17769
     2        HIEGHT                            A(3)=.657319
     3        WIDTH
```

Once all of the Initial Values have been entered, all the

Figure 3.1A - Sample Dialogue for Interactive Session information for the part attributes is displayed and you are asked if everything is correct.

```
Attribute #   Description                          Initial Value
     1        LENGTH                               A(1)=.03571

     2        HIEGHT                               A(2)=.17769

     3        WIDTH                                A(3)=.657319


  IS EVERYTHING CORRECT? (Y/N): N
```

If you answer 'N' the following dialogue occurs:

```
MODIFY ATTRIBUTE #: 2

ENTER THE DESCRIPTION : HEIGHT

ENTER THE INITIAL VALUE : A(2)=.17769

MODIFY ANOTHER ATTRIBUTE? (Y/N): N
```

If you type 'Y' and wish to modify another attribute, then the same dialogue will occur again. If you type 'N' then the attributes appear on the screen with all the changes made.

```
Attribute #   Description                          Initial Value
     1        LENGTH                               A(1)=.03571

     2        HEIGHT                               A(2)=.17769

     3        WIDTH                                A(3)=.657319


  IS EVERYTHING CORRECT? (Y/N): Y

  ADD ANOTHER ATTRIBUTE? (Y/N): Y
```

If you answer 'Y' and wish to add another attribute you are given the next attribute number as a prompt and the following appears:

```
ENTER INFORMATION FOR ATTRIBUTE #: 4
ENTER THE DESCRIPTION : WEIGHT
ENTER THE INITIAL VALUE : A(4)=12.5786
```

Once all the attributes have been added, they are listed and you are given the chance to modify them again.

Attribute #	Description	Initial Value
1	LENGTH	A(1)=.03571
2	HEIGHT	A(2)=.17769
3	WIDTH	A(3)=.657319
4	WEIGHT	A(4)=12.5786

```
IS EVERYTHING CORRECT? (Y/N): Y
ADD ANOTHER ATTRIBUTE? (Y/N): N
```

If everything is correct and you do not wish to enter another attribute you will be given the option of entering additional initial equations. These equations are used to further describe the system and can be any valid FORTRAN statement. The variable XX(I) is defined as a global variable that may be used in these equations.

If you type 'Y' to enter additional initial equations the following screen dialogue occurs:

```
Do you want to enter Additional Initial Equations? (Y/N): Y
ENTER EQUATION  1
XX(1)+SIN(A(1))

XX(1)+SIN(A(1))
IS THIS CORRECT? (Y/N): N
```

If it is not correct type 'N', and re-enter the equation.

```
ENTER THE CORRECTED EQUATION
XX(1)=SIN(A(1))

XX(1)=SIN(A(1))
IS THIS CORRECT? (Y/N): Y

ENTER ANOTHER EQUATION? (Y/N): N
```

If it is correct type 'Y' and enter all other initial equations.

Enter the Processes

You must now enter the processes. You are given the process number as a prompt and asked to enter a description of each. The screen appears as follows:

```
ENTER THE DESCRIPTION OF EACH PROCESS

PROCESS #  DESCRIPTION
   1       MOLD THE PART
   2       BORE HOLE THROUGH CENTER
   3       DONE
```

Once you have finished entering all the process descriptions, type the word "DONE" as the next decription. All of the process numbers and descriptions will then be displayed to insure that they are correct.

```
PROCESS #  DESCRIPTION
   1       MOLD THE PART
   2       BORE HOLE THROUGH CENTER

IS EVERYTHING CORRECT? (Y/N): N
```

If an 'N' is entered the following dialogue occurs:

```
ENTER THE # OF THE PROCESS TO BE MODIFIED: 1

ENTER THE DESCRIPTION: SHAPE THE PART
```

The processes and their descriptions are displayed again and you have the option of correcting another error.

```
PROCESS #  DESCRIPTION
    1      SHAPE THE PART
    2      BORE HOLE THROUGH CENTER
```

```
IS EVERYTHING CORRECT? (Y/N): Y
```

Enter 'Y' once all errors have been eliminated. You will then see the following question:

```
ADD ANOTHER PROCESS? (Y/N):
```

If you wish to add another process type 'Y'. The following prompt will appear with the next process number.

```
ENTER THE DESCRIPTION FOR PROCESS 3
HEAT THE PART
```

Once the description is entered, all the processes are displayed so you may check for errors.

```
PROCESS #  DESCRIPTION
    1      SHAPE THE PART
    2      BORE HOLE THROUGH CENTER
    3      HEAT THE PART
```

```
IS EVERYTHING CORRECT? (Y/N): Y

ADD ANOTHER PROCESS? (Y/N):N
```

After all processes have been entered and all errors have been corrected, enter 'Y' and 'N', respectively, to the above questions.

It is now time to determine which attributes are modified by each process. The following is an example of the dialogue which typically occurs.

```
DETERMINE WHICH ATTRIBUTES ARE MODIFIED BY
PROCESS  1 SHAPE THE PART

DOES PROCESS  1 MODIFY                        ? (Y/N): Y
ATTRIBUTE 1 LENGTH
```

If you answer 'N', then the following will appear:

```
DOES PROCESS  1 MODIFY                        ? (Y/N):
ATTRIBUTE 2 HEIGHT
```

Each time an 'N' is entered, you will be asked if the next attribute is modified by that process.

If you enter 'Y', the following information is entered. Be sure to include a decimal point when entering the tolerances, cell width and upper limit on the first cell. The number of cells, however must be an integer.

```
ENTER THE MODIFYING EQUATION

A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(2.1,.02))

ENTER THE LOWER TOLERANCE .1234

ENTER THE UPPER TOLERANCE .7984
```

```
REJECT PARTS OUT OF TOLENANCE? (Y/N): N

COLLECT HISTOGRAM DATA? (Y/N): Y

ENTER THE # OF CELLS: 7

ENTER THE CELL WIDTH: .27597

ENTER THE UPPER LIMIT OF THE FIRST CELL: .00034

DO YOU WANT TO ENTER ADDITIONAL EQUATIONS? (Y/N): Y

ENTER THE EQUATION

XX(3)=COS(A(2))


IS THIS CORRECT? (Y/N): N

ENTER THE CORRECTED EQUATION

XX(4)=RNORM(3.45,.98)

XX(4)=RNORM(3.45,.98)

IS THIS CORRECT? (Y/N): Y

ENTER ANOTHER EQUATION? (Y/N): N
```

This information is entered for each attribute that is modified by a particular process.

Once all information, for all the attributes modified by this process, is entered it is listed on the screen as follows:

```
ATTRIBUTES FOR PROCESS   1

ATTRIBUTE DESCRIPTION        TOL1        TOL2         HIST REJECT
    1      LENGTH            .1234       .7984         H
    2      HEIGHT            1.324       4.9734        H       R

MODIFYING EQUATION
A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(2.1,.02))
A(2)=4*ABS(XX(3))+SQRT(XX(4))


IS EVERYTHING CORRECT? (Y/N): N
```

If you answer 'N' then the following appears:

```
MODIFY ATTRIBUTE # : 1
```

You are then asked to enter all the attribute information as before. Once this is done, the data is displayed on the screen.

```
ATTRIBUTES FOR PROCESS   1

ATTRIBUTE DESCRIPTION        TOL1        TOL2         HIST REJECT
    1      LENGTH            .1234       .7984         H
    2      HEIGHT            1.324       4.9734        H       R

MODIFYING EQUATION
A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(4.6,.09))
A(2)=4*ABS(XX(3))+SQRT(XX(4))


IS EVERYTHING CORRECT? (Y/N): Y
```

If you answer 'Y' and histogram specifications were entered for one or more of the attributes, the following appears on the screen:

HISTOGRAM SPECIFICATIONS FOR THE ATTRIBUTES OF PROCESS 1

ATTRIBUTE	DESCRIPTION	# CELLS	CELL WIDTH	UPPER LIMIT
1	LENGTH	7	.454300	.003400
2	HEIGHT	8	1.313810	.036250

IS EVERYTHING CORRECT? (Y/N): N

If 'N' is entered , you will see the following:

MODIFY ATTRIBUTE # : 1

You are given the option of entering histogram data for that attribute. If you enter 'Y', data is entered as before. All attributes are listed again and you are given the option of making corrections. Once all corrections have been made, you will be asked:

DO YOU WANT TO ADD ANOTHER ATTRIBUTE? (Y/N): N

If you answer 'Y', you will be asked to enter all the information for the attribute you selected. If you answer 'N', you will be asked to determine which attributes are modified by the remaining processes.

Once it has been determined which attributes are modified by all processes the following will appear:

```
ENTER THE NUMBER OF PARTS TO BE PROCESSED: 300

300 PARTS
IS THIS CORRECT? (Y/N): N
```

If you enter 'N', you will again be asked to enter the number of parts.

These options provide the user with the ability to modify any part of the input file. For example, if the user wished to modify an attribute of a particular process, he would enter 1. The processes and their descriptions would then be listed and the user would choose the process he wanted to modify. The following would then be presented:

1) Modify the Process Description
2) Modify the Attributes
Enter (1 or 2):

The user would enter 2 since he wants to modify an attribute. Once he did so, the following menu would appear on the screen:

1) Add an Attribute
2) Modify an Attribute
3) Delete an Attribute
4) Modify Additional Equations
Enter (1,2,3 or 4):

At this point, the user is provided with the means to modify any attribute of the process previously entered.

While an old file is being modified, checks are constantly being made to insure that the data is being entered correctly. This eliminates the tedious chore of re-entering all previous data if just one mistake is made.

When the user is either finished entering data for a new file or finished modifying an old file, the information is saved under the file name specified by the user.

3.2 Input File

The input file, which is constructed during the interactive session, consists of a list of all the processes in the production system which alter one or more of the dimensions of the part being manufactured. Histogram specifications and the number of parts to be processed are also included.

A part can be best described as an array containing a list of attributes which are to be modified and a process as a list of the attributes that the machine must alter. The specific modification is expressed by a distribution and a part is either accepted or rejected based on the tolerance given. The user has the option of removing the part from the line or to let it continue processing even when out of tolerance on a particular attribute. Data for a histogram on the range of the values of a modified attribute is collected for a particular attribute if so specified by the user. The stopping rule is used to determine when enough parts have been manufactured and the simluation should be haulted. A sample input file can be seen in Figure 3.2A.

A creation program, "CREATPRO", is implemented to convert part of the input file into a FORTRAN subroutine called subroutine PROCESS. This code, which represents the processes and the attributes they modify, is used by the simulation to model the production system. An example of subroutine PROCESS is provided in Figure 3.2B. Each time the subroutine is called a specified section of the code is executed; each section represents a process. The argument I indicates which particular process is to be employed. Once this is determined, all the code, between the corresponding CONTINUE and RETURN statements, is executed. This code consists of the modifying equations for the attributes altered by that process, conditional IF statements for the specified tolerances with corresponding counters for the number of parts out of tolerance and any additional equations the user may have entered. After the simulation is created it is compiled and linked with the simulaion program.

The creation program also generates a condensed version of the input file. This scratch file is exactly the same as the input file except that it does not contain any modifying or additional equations. An example of this file can be

```
 3 4 2 4
 INITIALIZE ATTRIBUTES
 1 LENGTH
A(1)=.03571
XX(1)=SIN(A(1))
 2 HEIGHT
A(2)=.17769
 3 WIDTH
A(3)=.657319
 4 WEIGHT
A(4)=12.5786
 SHAPE THE PART
 1 LENGTH                                   .1234       .7984       H
A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(4.6,.09)))
XX(4)=RNORM(3.495,.78)
XX(3)=COS(A(3))
 2 HEIGHT                                   1.324       4.9734      H R
A(2)=4*ABS(XX(3))+SQRT(XX(4))
 BORE HOLE THROUGH CENTER
 1 LENGTH                                   .00253      .06157        R
A(1)=SIN(A(1))**2
 2 HEIGHT                                   .00796      1.99736     H
A(2)=ABS(COS(A(3)))+RNORM(3.29,2.4)
 3 WIDTH                                    .00534      .71998      H R
A(3)=SQRT(A(1))
 4 WEIGHT                                   .00318      5.3432
A(4)=ABS(COS(A(2)))+RNORM(5.4,.3)
 7     .043000     .000340
 8     .180000    3.65100
 6     .063000    1.09890
 9     .019500     .000027
 325
```

Figure 3.2A

Sample Input File

```
      SUBROUTINE PROCESS(I)
      COMMON/SIMCOM/A(5),NA(5),NX(10,5),KX(10,5),NPARTS(10),
     + XMEAN(10,5),XSTD(10,5),NATT(10,5),NCELLS(10,5),
     + CWIDTH(10,5),UPLIM(10,5),KCELL(10,5,11),NPROC,
     + DESCPROC,DESCATT,HIST,XX(10),REJ,TOL1,TOL2,
     + DMEAN(10,5),DSTD(10,5)
      CHARACTER DESCPROC(10)*66,DESCATT(10,5)*30,HIST(10,5)*1,
     + REJ(10,5)*1,TOL1(10,5)*10,TOL2(10,5)*10
      GO TO ( 1, 2, 3)I
1     CONTINUE
      A(1)=.03571
      XX(1)=SIN(A(1))
      A(2)=.17769
      A(3)=.657319
      A(4)=12.5786
      RETURN
2     CONTINUE
      A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(4.6,.09)))
      XX(4)=RNORM(3.495,.78)
      XX(3)=COS(A(3))
      IF(A( 1).LT..1234      )NX( 2,  1)=NX( 2,  1)+1
      IF(A( 1).GT..7984      )NX( 2,  1)=NX( 2,  1)+1
      A(2)=4*ABS(XX(3))+SQRT(XX(4))
      IF(A( 2).LT.1.324      )NX( 2,  2)=NX( 2,  2)+1
      IF(A( 2).GT.4.9734     )NX( 2,  2)=NX( 2,  2)+1
      RETURN
3     CONTINUE
      A(1)=SIN(A(1))**2
      IF(A( 1).LT..00253     )NX( 3,  1)=NX( 3,  1)+1
      IF(A( 1).GT..06157     )NX( 3,  1)=NX( 3,  1)+1
      A(2)=ABS(COS(A(3)))+RNORM(3.29,2.4)
      IF(A( 2).LT..00796     )NX( 3,  2)=NX( 3,  2)+1
      IF(A( 2).GT.1.99736    )NX( 3,  2)=NX( 3,  2)+1
      A(3)=SQRT(A(1))
      IF(A( 3).LT..00534     )NX( 3,  3)=NX( 3,  3)+1
      IF(A( 3).GT..71998     )NX( 3,  3)=NX( 3,  3)+1
      A(4)=ABS(COS(A(2)))+RNORM(5.4,.3)
      IF(A( 4).LT..00318     )NX( 3,  4)=NX( 3,  4)+1
      IF(A( 4).GT.5.3432     )NX( 3,  4)=NX( 3,  4)+1
      RETURN
      END
```

Figure 3.2B

Sample of Subroutine PROCESS

seen in Figure 3.2C. The simuation uses this file to determine which attributes are to have histogram data collected and at which processes parts are to be taken off the line if they are out of tolerance.

3.3 Simulation

In order to run the simulation, the program name "SIMULATE" is entered. After a few seconds the user is presented with the following options:

1. Use an OLD Output File as a NEW Output File
2. Create a NEW Output File Name
3. List all Old Output File Names

Enter (1,2 or 3):

A name, for the output file created by the simulation, must be specified. If the user decides to use an old name, all the data in the old file will be lost. The same guidelines must be followed when entering the output file name as when entering the input file name. Checks are made to insure that the name adheres to these guidelines and if it does not an error message is given. If an old file name is entered and it does not exist on the old files list, an error message is displayed. If a new file name is entered and it

```
 3 4 2 4
INITIALIZE ATTRIBUTES
 1 LENGTH
 2 HEIGHT
 3 WIDTH
 4 WEIGHT
SHAPE THE PART
 1 LENGTH                              .1234       .7984       H
 2 HEIGHT                              1.324       4.9734      H R
BORE HOLE THROUGH CENTER
 1 LENGTH                              .00253      .06157        R
 2 HEIGHT                              .00796      1.99736     H
 3 WIDTH                               .00534      .71998      H R
 4 WEIGHT                              .00318      5.3432
 7     .043000     .000340
 8     .180000    3.651000
 6     .063000    1.098900
 9     .019500     .000027
 325
```

Figure 3.2C

Sample Scratch File

does exist on the old files list, an error message is also displayed.

After the file name is determined, a message, which indicates that the simulation is running, appears on the screen for a few seconds. The user will then be furnished with in-progress reports showing the number of parts already processed and the total number of rejects. The mean and standard deviation of the attributes being modified are automatically determined for each process. Histogram data is collected for the processes and attributes which were specified by the user during the interactive session. Once the simulation is finished, all information genertated is saved in the user specified ouptut file.

3.4 Output File

The output file, which is generated by the simulation, contains the following information:

1) the number of processes

2) the number of attributes modified by each process

3) the number of parts processed by each process

4) a list of the attributes modified by each process, including the mean and standard deviation, the difference of means and standard deviations and the number of parts out of tolerance

5) if histogram data was collected for an attribute,

the number of cells, the cell width and the upper limit on the first cell are listed along with the number of parts in each cell

A sample output file is provided in Figure 3.4A.

3.5 Statistical Reports

Results of the simulation can be reviewed by executing the program "SUMMARY". The output file generated by the simulation is used to construct summary reports and histogram tables. Once the user enters the name of the output file to be used and reads the messages displayed, the following options appear:

1. Display Summary Reports
2. Display Histogram Data
3. End the Program

Enter (1,2 or 3):

Whether the user chooses to see summary reports or histogram data, he is able to construct the report to fit his particular needs. The user may see a report for all the data collected, for all the data for a praticular process, for all the data for a specific attribute or for all the data for a particular process with a specific attribute. These reports may be displayed on the screen and/or printed

```
3 4 2 4
325INITIALIZE ATTRIBUTES
1 LENGTH
  .035710    .000001    .000000    .000000    0
2 HEIGHT
  .177689    .000019    .000000    .000000    0
3 WIDTH
  .657318    .000046    .000000    .000000    0
4 WEIGHT
12.578610    .000237    .000000    .000000    0
325SHAPE THE PART
1 LENGTH                              .1234      .7984       H
  .552311    .013986    .516600    .013986   89
7    .043000    .000340
  0     0     5    15    15    12     9   269
2 HEIGHT                              1.324      4.9734      H R
 3.828693    .013893   3.651003    .013892    0
8    .180000   3.651000
 83    58    83   101     0     0     0     0     0
325BORE HOLE THROUGH CENTER
1 LENGTH                              .00253     .06157        R
  .305453    .011194   -.246857    .003299  270
2 HEIGHT                              .00796     1.99736     H
 1.294890    .016017  -2.533801    .020741    0
6    .063000   1.098900
103    22    17    15    16    30   122
3 WIDTH                               .00534     .71998      H R
  .508956    .011988   -.148363    .011988   70
9    .019500    .000027
  0     0     0     0     1     6    10     7     7   294
4 WEIGHT                              .00318     5.3432
  .811413    .020086 -11.767190    .020082    0
```

Figure 3.4A

Sample Output File

on the printer. Figures 3.5A - 3.5H contain samples of the various types of output reports that may be generated.

****
** SIMULATION SUMMARY REPORT **
****

PROCESS 1 SHAPE THE PART

ATTRIBUTES MODIFIED BY PROCESS 1

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	89
2	HEIGHT	325	0

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.552311	.013986	.035710	.000001	.516600	.013986
2	3.828693	.013893	.177689	.000019	3.651003	.013892

PROCESS 2 BORE HOLE THROUGH CENTER

ATTRIBUTES MODIFIED BY PROCESS 2

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	270
2	HEIGHT	325	0
3	WIDTH	325	70
4	WEIGHT	325	0

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.305453	.011194	.552311	.013986	-.246857	.003299
2	1.294890	.016017	3.828693	.013893	-2.533801	.020741
3	.508956	.011988	.657318	.000046	-.148363	.011988
4	.811413	.020086	12.578610	.000237	-11.767190	.020082

Figure 3.5A - Entire Summary Report

```
*******************************
** SIMULATION SUMMARY REPORT **
*******************************
```

PROCESS 1 SHAPE THE PART

ATTRIBUTES MODIFIED BY PROCESS 1

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	89

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.552311	.013986	.035710	.000001	.516600	.013986

PROCESS 2 BORE HOLE THROUGH CENTER

ATTRIBUTES MODIFIED BY PROCESS 2

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	270

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.305453	.011194	.552311	.013986	-.246857	.003299

Figure 3.5B - Summary Report of a Particular Attribute

```
******************************
** SIMULATION SUMMARY REPORT **
******************************
```

PROCESS 2 BORE HOLE THROUGH CENTER

ATTRIBUTES MODIFIED BY PROCESS 2

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	270
2	HEIGHT	325	0
3	WIDTH	325	70
4	WEIGHT	325	0

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.305453	.011194	.552311	.013986	-.246857	.003299
2	1.294890	.016017	3.828693	.013893	-2.533801	.020741
3	.508956	.011988	.657318	.000046	-.148363	.011988
4	.811413	.020086	12.578610	.000237	-11.767190	.020082

Figure 3.5C - Summary Report of a Particular Process

****
** SIMULATION SUMMARY REPORT **
****

PROCESS 2 BORE HOLE THROUGH CENTER

ATTRIBUTES MODIFIED BY PROCESS 2

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	270

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.305453	.011194	.552311	.013986	-.246857	.003299

Figure 3.5D - Summary Report of a Particular Process with a Particular Attribute

HISTOGRAM DATA FOR
PROCESS 1 ATTRIBUTE 1: LENGTH

OBSERVED NO. PARTS	RELATIVE NO. PARTS	CUMULATIVE NO. PARTS	UPPER CELL LIMIT
0	.000000	.000000	.000340
0	.000000	.000000	.043340
5	.015385	.015385	.086340
15	.046154	.061538	.129340
15	.046154	.107692	.172340
12	.036923	.144615	.215340
9	.027692	.172308	.258340
269	.827692	1.000000	INFINITY

325			

Figure 3.5E

Sample Histogram Table

HISTOGRAM DATA FOR
PROCESS 1 ATTRIBUTE 2: HEIGHT

OBSERVED NO. PARTS	RELATIVE NO. PARTS	CUMULATIVE NO. PARTS	UPPER CELL LIMIT
83	.255385	.255385	3.651000
58	.178462	.433846	3.831000
83	.255385	.689231	4.011000
101	.310769	1.000000	4.191000
0	.000000	1.000000	4.371000
0	.000000	1.000000	4.551000
0	.000000	1.000000	4.730999
0	.000000	1.000000	4.910999
0	.000000	1.000000	INFINITY

325			

Figure 3.5F

Sample Histogram Table

HISTOGRAM DATA FOR
PROCESS 2 ATTRIBUTE 2: HEIGHT

OBSERVED NO. PARTS	RELATIVE NO. PARTS	CUMULATIVE NO. PARTS	UPPER CELL LIMIT
103	.316923	.316923	1.098900
22	.067692	.384615	1.161900
17	.052308	.436923	1.224900
15	.046154	.483077	1.287900
16	.049231	.532308	1.350900
30	.092308	.624615	1.413900
122	.375385	1.000000	INFINITY
325			

Figure 3.5G

Sample Histogram Table

HISTOGRAM DATA FOR
PROCESS 2 ATTRIBUTE 3: WIDTH

OBSERVED NO. PARTS	RELATIVE NO. PARTS	CUMULATIVE NO. PARTS	UPPER CELL LIMIT
0	.000000	.000000	.000027
0	.000000	.000000	.019527
0	.000000	.000000	.039027
0	.000000	.000000	.058527
1	.003077	.003077	.078027
6	.018462	.021538	.097527
10	.030769	.052308	.117027
7	.021538	.073846	.136527
7	.021538	.095385	.156027
294	.904615	1.000000	INFINITY

325			

Figure 3.5H

Sample Histogram Table

4. Program Module Descriptions

The entire project is broken up into four major programs. Each program in turn is divided into numerous small modules or subroutines, each of which is designed for a specific purpose or task. This provides greater control over the situation while creating and modifying each program since only small portions must be dealt with at any one time. Descriptions of all the various types of procedures incorporated into the four programs follow. A complete listing of all the FORTRAN code can be found in the office of the Department of Industrial Engineering at Lehigh University.

4.1 The first program, PUTINFIL, is used to gererate the input file needed to run the simulation. The same program is used to modify an input file which already exists.

Program PUTINFIL- This is the main program which calls all appropriate subroutines to either enter a new file or modify an existing one. It is also used to open all I/O and data files. A value of one is sent to subroutine IOFILE to indicate an input file is to be used. If the I returned from subroutine IOFILE is equal to one then an existing file is to be modified, however, if the I is equal to two a new file is to be created. If an existing input file is to be used exactly the way it is, one would answer no to the question "modify the file?".

1) Subroutine TITLE- This subroutine writes the program title, version number and authors' names to the screen.

2) Subroutine REMIND- This subroutine displays special reminders about ways to exit the program, list instructions or continue a list.

3) Subroutine INPINIT(OPT)- This subroutine initializes all variables and arrays used by the program. OPT is used as the option from subroutine ESCAPE and is also initialized to a blank.

4) Subroutine IOFILE(OPT,I,KF)- This subroutine determines the name of the input/output file. An old file may be used, a new file created or a list of all old files can be displayed. If a new file name is entered subroutine CHECK is called to determine if all file naming specifications have been met. The subroutine returns with the name of the file to be used stored in variable INP. The arguments are defined as follows. If KF is entered equal to one an input

file name is to be determined. If Kf is entered equal to two an output file name is to be determined. OPT is the option used by subroutine ESCAPE and is returned as S to start over or Q to quit the program. I is returned as one if an existing file is to be modified and as two if a new file is to be created.

5) Subroutine BLANK- This subroutine is used to clear the screen by implementing Halo Graphics.

6) Subroutine ESCAPE(OPT)- This subroutine is called to either get help, start over or quit the program. If help is needed subroutine INSTRUCT is called. OPT is returned as C to continue the program from where "ESC" was initiated, S to start the program over or Q to quit the program.

7) Subroutine INSTRUCT- This subroutine reads file HELP, which contains program instructions, and displays it on the screen line by line. A counter is used to insure that only twenty lines are listed at a time. A carriage return is needed to continue displaying the file once it has been stoped.

8) Subroutine LFILES(KF)- Subroutine LFILES lists all the Input/output files which already exist. If KF is entered as

one, input files are listed. If KF is entered as two, output files are listed. The file names are stored in a file and a counter is used to insure that only twenty lines are listed at a time. A carriage return is need to continue displaying the file once it is haulted.

9) Subroutine OLDFIL(IFILE)- This subroutine determines the name of the existing file to be used as input/output. It checks to make sure the file actually exists. If the file does not exist an error message is displayed and IFILE is returned with a value N. If the file does exist the file name is returned in IFILE.

10) Subroutine NEWFIL(KFILE,KK)- This subroutine determines a new file name and checks to see if the file already exists. If it doesn't, KFILE is returned as the name entered. If the file does exist, the option of using it is given. If the user decides not to use it KFILE is returned as N. If the file is to be used regardless of the fact that it already exists, the old file will be destroyed and a new one created. KFILE is returned as the name entered and KK is set equal to one to indicate an old file name is being used, otherwise KK is returned as zero.

11) Subroutine CHECK(KFILE)- This subroutine is used to

verify that file name KFILE conforms to the restrictions given in subroutine NEWFIL. If the name begins with a number or contains a special character an appropriate error message is displayed and the file name KFILE is set equal to N to indicate the file name contained an error.

12) Subroutine COPYINP- Subroutine COPYINP copies all the information from an existing input file into the appropriate variable and array names.

13) Subroutine PART- This subroutine determines the description of the attributes of the part and their initial values. The user is given an attribute number and responds with a description. Once all the descriptions have been entered the user types "DONE". The attribute number, description and a prompt A(#)= are displayed. As soon as all the initial values have been entered all the information is listed on the screen to determine if anything needs to be modified. A counter is used to insure that no more than twenty lines will be displayed at a time. A carriage return is needed to continue listing the information. If an attribute must be modified subroutine MODPART is called. Additional initial equations may also be added and are entered by calling subroutine EXTRAIPE.

14) Subroutine MODPART- This subroutine is called to modify the description and/or the initial value of the part attribute. A list of the attributes along with their descriptions and initial values is already present. More than one attribute may be modified.

15) Subroutine ADDPART(KB)- Subroutine ADDPART determines if another initial attribute may be entered by making sure NA(1), the number of attributes modified by process 1 (process 0 to the user), is less than the maximum allowable number of attributes. If this condition holds and the user wishes to enter another attribute, he is given prompts for the information, the counter for the number of initial attributes is increased and KB is returned set equal to one indicating that an additional attibute was entered.

16) Subroutine EXTRAIPE- This subroutine is called to enter additional intial equations. It starts with the first available place MODEQN(1,1,2) which is process 1, attribute 1, equation 2 and allows the user to enter the equations until the last place has been occupied, MODEQN(1,NA(1),#), which is process 1, last attribute, largest number allowed determined by the dimension statement. These equations are placed, in the order entered, between the initial value equations in subroutine PROCESS, (see note below). Once an

equation is entered, it is displayed on the screen to insure that it is correct. A message is displayed if no more room is available for another initial additional equation.

NOTE: The additional initial equations are placed in subroutine PROCESS as follows:

The array MODEQN is dimensioned as MODEQN(I,J,N)

```
A(1)=some value
MODEQN(1,1,2)
    .
    .
MODEQN(1,1,N)
A(2)=some value
MODEQN(1,2,2)
    .
    .
MODEQN(1,2,N)
A(NA(1))=some value
MODEQN(1,NA(1),2)
    .
    .
MODEQN(1,NA(1),N)
```

The equations are placed in order starting with the first place under attribute 1. Additional equations for processes other than process 1 (process 0 to the user) are not stored in this manner.

17) Subroutine STATION- This subroutine determines the number of processes in the system and the description of each. It prompts the user with a process number and then waits for a description to be entered. When finished, the user responds with "DONE". A counter is used to calculate the actual number of processes; this number is stored in NPROC. Subroutine CHPRO is called to check the process description for errors and then subroutine ENTERATT is

called to determine which attributes are modified by the processes defined. Subroutine STATION also determines the number of parts that are to be processed.

18) Subroutine CHPRO- This subroutine verifies that all processes and the descriptions entered are correct. If one must be modified it asks for the process number and then the corrected description. If another process is to be entered, subroutine ADDPRO is called with the number of the process to be entered as an argument.

19) Subroutine ADDPRO(K)- Subroutine ADDPRO is called so another process and description may be entered. The counter for the number of processes is increased and a check is made to be sure there is room for another process. The argument K is the number of the new process.

20) Subroutine ENTERATT(K)- This subroutine is used to specify which attributes are modified by a particular process. The argument K is equal to the process in question. The program looks at all the attributes defined by subroutine PART and determines which ones are modified by process K. When a particular attribute is chosen, the description is copied from the part attribute and the attribute number is set equal to the part attribute number.

Subroutine ATTRIB is then called to enter more information regarding the attribute chosen. Argument KH is set equal to zero which indicates that all information must be added. If the user wants to enter additional equations under this particular attribute, subroutine ADDEQNS is called with the process number and attribute number as arguments. NA(K), the number of attributes modified by process K, is set equal to the number of attributes once they have all been entered. Then subroutine CHATT is called which is used to check all the data that has been collected for the attributes. Argument KSL is equal to one which indicates that no attributes are to be deleted. Argument K is equal to the process number. If NA(K) is less than the maximum allowable amount then another attribute may be added. Subroutine ADDATT is called for this purpose. If the value of NK is returned as one, an attribute has been added.

21) Subroutine ATTRIB(K,L,KH)- This subroutine collects all the data for attribute L which is modified by process K. It determines the modifying equation, lower tolerance and upper tolerance. It also determines if parts out of tolerance should be rejected and whether histogram data is to be collected. The number of cells, the cell width and the upper limit of the first cell are entered if appropriate. If the argument KH is entered equal to zero then all

information is to be collected. If KH is one, only the histogram specifications are to be entered.

22) Subroutine ADDEQNS(K,L)- This subroutine is used to enter additional equations under a particular attribute. It first checks to make sure there is an available location for the equation, if a space is not available an error message is printed. Once the equation is entered it is displayed so the user can check for errors. If there is an error it can be modified immediately. Argument K is equal to the process number and argument L is equal to the attribute number.

23) Subroutine CHATT(K,KSL)-Subroutine CHATT provides a check for all the attribute data entered for process K. All the data for all attributes modified by a particular process are displayed on the screen. Any data that is incorrect may be modified at this time. If KSL is entered equal to zero, an attribute may be deleted. If an attribute is to be modified subroutine ATTRIB is called with KH equal to zero which indicates that all information may be modified. Once all errors are removed, the histogram specifications are displayed. If an error exists subroutine ATTRIB is called with KH equal to one which indicates that only histogram information is to be modified. This entire process is repeated until there are no errors.

24) Subroutine DELATT(I)- This subroutine deletes one of the attributes thought to be modified by a particular process. The argument I is the process number to the user, the program stores the actual process as I+1. All the variables associated with process I+1 and the attribute to be deleted are reinitialized and NA(I+1), the number of attributes modified by process I+1, is decreased by one.

25) Subroutine ADDATT(NK,K)- This subroutine is called when another attribute is to be added. It determines which attributes have already been modified by process K and then determines which of the attributes not yet modified is to be added. NA(K), the number of attributes modified by process K, is increased by one, the description for the newly chosen attribute is copied from the corresponding part attribute description and the number of the attribute is set according to the attribute's array position. Subroutine ATTRIB is then called with KH equal to zero. If additional equations are to be entered under this particular attribute, subroutine ADDEQNS is called. Subroutine ADDATT is repeated until either the user does not want to add any more attributes or until no more may be added. Argument NK is returned equal to one if an attribute was added.

26) Subroutine MODIFY(OPT)- This subroutine determines

whether a process is to be modified, added or deleted or if the number of parts to be processed is to be changed. If an "ESC" is entered rather than one of the five options, subroutine ESCAPE is called and returns a value in OPT. If a process is to be modified subroutine MODPRO is called, if a process is to be added subroutine INSPRO is called and if a process is to be deleted subroutine DELPRO is called.

27) Subroutine MODPRO(OPT)- This subroutine is called to modify a process. The processes and their descriptions are listed on the screen. A process number is then entered as the one to be modified. The user is then given a choice between modifying the process description or modifying the attributes. If the process description is to be corrected, the user is given a prompt to enter a new description and a check for an error is made immediately. If the attributes are to be modified, subroutine MODATT is called with I equal to the process number entered by the user.

28) Subroutine MODATT(OPT,I)- Subroutine MODATT displays four options, add an attribute, modify an attribte, delete an attribute and modify additional equations. If "ESC" is entered subroutine ESCAPE is called. If argument I is entered as zero, then subroutine MODINIT is called since initial attributes are to be modified. For all other values

of I, subroutine MODOTHER is called. The argument J for both of these subroutines refers to the choice made in this routine.

29) Subroutine MODINIT(J,I)- This subroutine modifies the attributes of process 1 (process 0 to the user). If J is equal to one subroutine ADDPART is called to add another attribute. If KB is returned equal to one then an attribute was added so subroutine CHPART is called, with L equal to one, to check all data entered. If J equals two, an attribute is to be modified, subroutine CHPART is called with L equal to two. If J is equal to three, an attribute is to be deleted, subroutine CHPART is called with L equal to three. Finally , if J equals four, additional equations are to be modified, subroutine PEQNS is called. Argument I is equal to zero and is actually representing process 1 which is process 0 to the user.

30) Subroutine CHPART(L,I)- This subroutine displays the attribute description and initial value for each attribute modified by process 1. If L is equal to two subroutine MODPART is called to modify an attribute. If L is equal to three subroutine DELATT is called to delete an attribute. If L is equal to one the user checks for errors and if an error exists subroutine MODPART is called to correct it.

31) Subroutine PEQNS(I)- Subroutine PEQNS is called in order to modify, add or delete additional equations. The program determines which attributes are modified by process I+1. If a particular attribute has no additional equations under it, an option is given to add them and subroutine ADDEQNS is called. If the attribute already had additional equations under it, they are listed on the screen. Then the user is given the option to delete and equation, add an equation or modify an equation. If an equation is to be deleted subroutine DELEQN is called with argument L equal to the process number and M equal to the attribute number. If an equation is to be added, the program asks for the equation and then displays the new list. If an equation is to be modified, the user enters the number of the equation and the corrected version. All equations are then relisted and the options are displayed again.

32) Subroutine DELEQN(L,M)- Subroutine DELEQN is called to delete an additional equation. Argument L is the process number and M is the attribute number. The program determines the number of the equation to be deleted and sets it equal to a blank. Then all other equations below it are moved up to eliminate the gap.

33) Subroutine MODOTHER(I,J)- This subroutine modifies the attributes of all processes except process 1. Argument I+1 is the actual process number and J is the number of the option chosen. If J is equal to one, add an attribute, subroutine ADDATT is called. If NK is returned equal to one, and attribute was added and the information added must be checked by subroutine CHATT. KSL is entered as zero so an attribute will not be deleted. If J is equal to two, modify an attribute, subroutine CHATT is called with KSL equal to one. If J is equal to three, delete an attribute, subroutine CHATT is called with KSL equal to zero so an attribute may be deleted. Finally if J is equal to four, modify an additional equation, subroutine PEQNS is called.

34) Subroutine INSPRO- This subroutine inserts a new process into the system. The program determines if another process can be added. If one can not be added an error message is printed. If one can be added the existing processes are listed and the user is asked to enter the number of the process which the new process is to follow. The process after the one indicated are shifted down in the arrays to make room for the new process. The variables to be used for the new process are then reinitialized and the new data is entered.

35) Subroutine DELPRO- Subroutine DELPRO deletes an entire process by shifting processes that come after the one to be deleted up one place. The variables are reinitialized as the data is moved. The process counter is also decreased by one.

36) Subroutine INFOCOPY- This subroutine writes all the information contained in all the variables and arrays to the user specified input file and to a file called "INPUT" which is used to create subroutine PROCESS. Details on subroutine PROCESS are discussed later. The information contained in the arrays and variables is entered interactively through the use of program PUTINFIL.

4.2 The second program, CREATPRO, uses the file generated by program PUTINFIL to create subroutine PROCESS which is used in the simulation and represents the actual production line.

Program CREATPRO- This is the main program which calls all appropriate subroutines used to read the input file produced by program PUTINFIL, to create a FORTRAN subroutine called PROCESS, and a condensed file called SCRATCH. Once created, subroutine PROCESS must be renamed PROCESS.FOR and then complied. The object file along with file SCRATCH are then used to run the simulation.

1) Subroutine WRITPRO- Subroutine WRITPRO initializes all the arrays and writes the first three lines of subroutine PROCESS to the appropriate file. It then uses the variable NPROC, the number of processes, to write the GO TO statement with the correct number of statement labels. Subroutine RWINPUT is then called to create the rest of the file (subroutine PROCESS).

2) Subroutine RWINPUT(NPROC)- This subroutine reads the number of attributes modified by each process from the file INPUT. It writes a labeled continue statement corresponding to each label in the GO TO statement. Subroutine READINP is called to read the information for process I and subroutine

WRITEINP is called to write the information for process I.

3) Subroutine READINP(I)- This subroutine reads all the information from file INPUT which pertains to process I.

4) Subroutine WRITEINP(I)- This subroutine writes the modifying equation and any additional equations corresponding to process I to the file for subroutine PROCESS. If tolerance values exist they are written to the file using the appropriate "IF" statement, a "GT" condition for upper tolerances and a "LT" condition for lower tolerances. When the "IF" statement is true a counter updates the number of rejects for that particular attribute of process I. This counter is written to the file as part of the "IF" statement.

5) Subroutine DUMPFILE(NPROC)- This subroutine writes a condensed version of the INPUT file to a file named SCRATCH. This file does not contain any of the modifying equations or additional equations.

4.3 The third program, SIMULATE, uses subroutine PROCESS and the SCRATCH file, generated by program CREATPRO, to run the actual simulation. The simulation runs for the specified number of parts and then produces an output file, the name of which is specified by the user. This provides the basis from which summary reports are constructed.

Program SIMULATE- This is the main program which calls all appropriate subroutines in order to run the simulation. Subroutine IOFILE determines the name of the user specified output file. KF is set with a value of two to indicate an output file is to be generated. As the simulation runs a count of the number of parts already processed and the number of parts rejected is displayed on the screen. Counters keep track of the number of parts actually processed by each station on the production line.

1) Subroutine SIMTITLE- Subroutine SIMTITLE produces the statement "Simulation is Running" in large pink letters.

2) Subroutine INITIAL- This subroutine initializes all the arrays used by the simulation. It also initializes the random number generator.

3) Subroutine READDATA(NRUNS)- This subroutine reads all

the data from the SCRATCH file produced by program CREATPRO. The argument NRUNS is returned as the number of parts to be processed.

4) Subroutine PROCESS(I)- This subroutine is created by program CREATPRO and is the actual representation of the production line. The GO TO statement determines which process the part must go through and is specified by the argument I. The IF statements, not including any additional equations that happen to be "IF", determine if a part is out of tolerance on a particular attibute and a counter is updated when the condition holds.

5) Function RNORM(XMN,ST)- Given values for XMN and ST, this function returns RNORM(XMN,ST) equal to a random number from a normal distribution with a mean equal to XMN and standard deviation equal to ST.

6) Function UNFRM(ULO,HI)- Given values for ULO and HI, this function returns UNFRM(ULO,HI) equal to a random number from a uniform distributions with ULO equal to the low end of the interval and HI equal to the high end.

7) Function TRIAG(TLO,TMD,THI)- Given values for TLO, TMD and THI, this function returns TRIAG(TLO,TMD,THI) equal to a

random number from a triangular distribution with TLO equal to the low end of the interval, THI equal to the high end and TMD equal to the mode.

8) Function EXPON(XMN)- Given a value for XMN, this function returns EXPON(XMN) equal to a random number from an exponential distribution with XMN equal to the mean.

9) Subroutine MEANSTD(I,AB)- This subroutine keeps track of values needed to calculate the mean and standard deviation for each attribute. The values needed to calculate the difference of the means and standard deviation are also stored. Argument I is equal to the process number and array AB contains the most recent update on the attribute specified.

10) Subroutine HISTOGRM(I)- Subroutine HISTOGRM determines which attributes of process I are to have histogram data collected. Once this is done it is determined, based on the histogram specifications given, which cell number should be increased by one indicating that the attribute dimension was within that cell's limits.

11) Subroutine OUTOFTOL(I,IFLAG)- This subroutine determines if any attribute of process I was out of tolerance by

checking the reject counters. IFLAG is returned with a value of one if an attribute was out of tolerance and the user specified that the part should be taken out of the production line.

12) Subroutine COMPMEAN- Subroutine COMPMEAN takes the values accumulated by subroutine MEANSTD and computes the actual means and standard deviations.

13) Subroutine INFOOUT- This subroutine writes all the data calculated and collected by the simulation to the user specified output file.

4.4 The fourth program, SUMMARY, provides output reports containing statistics, such as the mean and standard deviation, previous mean and standard deviation and the difference mean and standard deviation, for each attribute modified by each process. In addition the report indicates the number of parts that were out of tolerance for a particular attribute. The program also generates histogram data in the form of a table which lists frequency, relative frequency and cumulative frequency of the number of parts in a specific cell. All of these reports may be displayed on the screen and/or printed. It is also possible to just see small sections of a report.

Program SUMMARY- This is the main program which calls all the appropriate subroutines needed to generate the output reports. The user is given a choice between seeing summary reports or histogram reports.

1) Subroutine PICKOUT- This subroutine determines which output file is to be used to display reports. Subroutine OLDFIL is called if an output file is to be entered and subroutine LFILES is called with KF equal to two, to list all output files. KF equal to two indicates that output files are to be displayed.

2) Subroutine LETTERS- Subroutine LETTERS displays the title "Simulation Summary Reports" in large pink letters.

3) Subroutine SUMINIT- This subroutine is called to initialize all the arrays used by the program.

4) Subroutine SUMREAD- This subroutine reads the data from the user specified output file into the arrays used by the program.

5) Subroutine DMEANSTD- This subroutine determines the previous mean and standard deviation for each attribute. It locates the process, before the current process, where the attribute was last modified and assigns the previous mean and standard deviation of the attribute in question equal to the mean and standard deviation of the attribute when it was modified by the process located.

6) Subroutine NOTES- This subroutine displays a message to the user about how to exit the program by typing a ctrl-C.

7) Subroutine SUMCHOI- Subroutine SUMCHOI gives the option of displaying the entire summary report or displaying selected parts of the report. Subroutine SUMALL is called to see the whole report and subroutine SUMDETR is called to

see parts of it.

8) Subroutine SUMALL- This subroutine is called to generate entire summary reports. Printing and displaying option are given.

9) Subroutine SUMTYPE- This subroutine displays the entire summary report on the screen.

10) Subroutine SUMWRITE- This subroutine prints the entire summary report on the printer.

11) Subroutine SUMDETR- This subroutine is called to generate selected summary reports. Options on type of report are given.

12) Subroutine SUMATT- This subroutine displays and/or prints the report for a particular attribute. The report shows information for each time the attribute was modified.

13) Subroutine SUMTYA(NAT)- Subroutine SUMTYA displays a summary report, for attribute NAT, on the screen.

14) Subroutine SUMWRA(NAT)- Subroutine SUMWRA prints a summary report, for attribute NAT, on the printer.

15) Subroutine SUMPROC- This subroutine displays and/or prints the report for a particular process. The report shows information for each attribute modified by this process.

16) Subroutine SUMTP(NPR)- This subroutine displays a summary report for process NPR on the screen.

17) Subroutine SUMWRP(NPR)- This subroutine prints a summary report for process NPR on the printer.

18) Subroutine SUMSELC- This subroutine displays and prints a report for a particular process with a particular attribute. The subroutine also checks to make sure the process exists.

19) Subroutine SUMTAP(K,J)- This subroutine displays a report for process K with attribute J on the screen.

20) Subroutine SUMWAP(K,J)- This subroutine prints a report for process K with attribute J on the printer.

21) Subroutine HISTCHOI- This subroutine determines whether all histogram data or only selected histograms are to be displayed.

22) Subroutine HISTALL- Subroutine HISTALL prints and/or displays all histogram data.

23) Subroutine HISTDISP(I,J)- This subroutine displays histogram data for process I with attribute J on the screen.

24) Subroutine HISTTABP(I,J)- This subroutine prints histogram data for process I with attribute J on the printer.

25) Subroutine HISTDETR- Subroutine HISTDETR determines if histogram data it to be displayed for a particular attribute, a particular process or a particular process with a particular attribute.

26) Subroutine HISTSELC- This subroutine prints and displays histogram data for a particular process with a particular attribute. An error message is given if histogram data is not available for the requested process and attribute.

27) Subroutine HISTPROC- This subroutine prints and displays histogram data for a particular process.

28) Subroutine HISTATT- This subroutine prints and displays

all histogram data for a particular attribute. The data is displayed every time the attribute was modified and data was collected.

5. Areas of Further Development

There are three main areas in which further development of the model may be useful. The first is to incorporate additional stopping rules such as the following:

1) Run until a specified amount of raw material has been used

2) Run until a specified number of parts have been made or until there are a specified number of rejects

3) Run until a specified amount of raw material has been used or until there a specified number of rejects

4) Run until the variance of the mean of a particular attribute is within a certain interval

5) Run until the percent rejects (percent outside a specified interval) of a particular attribute is less than a certain amount

The second is to construct a simulation interrupt such that the user may stop the simulation and have all data collected up to that point saved in the output file. This device would prove to be a valuable time saver; since the user is able to monitor the simulation and he would be able to stop it and examine statistical reports, it may never have to run for as long as initially hypothesized. This would enable the user to not only save computer time, but time for himself as well.

The third area involves the development of graphical output for the histogram data collected by the simulation. If graphics were available the model would present a more complete report since the data would be represented in a pictorial as well as tabular form.

6. Variable Dictionary

A(I) = array of attributes being modified

AB(I,J) = value of attribute J after process I is done

CUM(K) = cumulative number of parts for cell K

CWIDTH(I,J) = cell width for the histogram for process I with attribute J

DESCATT(I,J) = description of attribute J in process I

DESCPROC(I) = description of process I

DMEAN(I,J) = difference of means for process I and attibute J

DSTD(I,J) = difference of standard deviations for process I with attribute J

FILE = string for a file name

HIST(I,J) = histogram flag for collecting data for process I with attribute J

ICR = value of a carriage return <cr>

IFILE = string for a file name

INP = name of the user specified input/output file

KCELL(I,J,K) = counters for the number of parts that fall within cell K on attribute J for process I

KFILE(I) = string for a file name

KP(I) = string variable

KX(I,J) = counter for the number of parts rejected by process I with attribute J before the current process is started

LBL = string variable used for the title, version number and authors' names

LINE = one line of instructions form file HELP

MESS = string variable used for simulation title and summary title

MODEQN(I,J,K) = modifying equation and additional equations for process J with attribute J, there may be up to K equations where the first MODEQN(I,J,1) is the modifying equation and all others are additional equations

NA(I) = number of attributes modified by process I

NATT(I,J) = number of the attribute for process I with attribute J, the number equals J

NCELLS(I,J) = number of cells for the histogram for process I with attribute J

NPARTS = number of parts to be processed, used in PUTINFIL

NPARTS(I) = number of parts processed by process I

NPROC = number of processes

NRUNS = number of parts to be processed

NX(I,J) = number of parts out of tolerance on attribute J at process I

NY = string used for yes 'Y' and no 'N' answers

OF = string for a file name

OPT = value returned from subroutine ESCAPE, S start over, Q quit, C continue

PMEAN(I,J) = previous mean of attribute J at process I

PSTD(I,J) = previous standard deviation of attribute J at process I

REJ(I,J) = rejection flag used to determine if a part is to be taken off the line if out of tolerance on attribute J at process I

REL(K) = relative number of parts in cell K

TEMP = temporary location for a modifying equation

TOL1(I,J) = lower tolerance for process I with attribute J

TOL2(I,J) = upper tolerance for Process I with attribute J

UPLIM(I,J) = upper limit on the first cell of the histogram for process I with attribute J

XMEAN(I,J) = mean value of attribute J at process I

XSTD(I,J) = standard deviation of the mean of attribute J at process I

XX(I) = global variable that may be used in subroutine PROCESS

Bibliography

IBM Disk Operating System Version 3.00. Boca Raton, Florida: International Business Machines Corporation, 1984.

IBM Guide to Operations Personal Computer XT. Boca Raton, Florida: International Business Machines Corporation, 1984.

Groover, Mikell P., Automation, Production Systems, and Computer-Aided Manufacturing. Englewood Cliff, New Jersey: Preintice-Hall, Inc., 1980.

Halo Graphics Reference Manual. New York: Media Sybernetics Inc., and Lifeboat Associates, 1984.

Microsoft FORTRAN Compiler for the MS-DOS Operating System. Bellevue, WA: Microsoft Corporation, 1984.

Rouse, Robert and Thomas Bugnitz, Programming the IBM Personal Computer: FORTRAN 77. New York: CBS College Publishing, Holt, Rinehart and Winston, 1983.

UnderWare, Inc., Brief, The Programmer's Editor. Norwell, MA: Solution Systems, 1984.

Appendix I

Appendix I contains a sample of subroutine PROCESS, copies of all the files created by the program, sample Summary Reports and Histogram Tables.

```
 3 4 2 4
 INITIALIZE ATTRIBUTES
 1 LENGTH
A(1)=.03571
XX(1)=SIN(A(1))
 2 HEIGHT
A(2)=.17769
 3 WIDTH
A(3)=.657319
 4 WEIGHT
A(4)=12.5786
 SHAPE THE PART
 1 LENGTH                                     .1234        .7984        H
A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(4.6,.09)))
XX(4)=RNORM(3.495,.78)
XX(3)=COS(A(3))
 2 HEIGHT                                     1.324        4.9734       H R
A(2)=4*ABS(XX(3))+SQRT(XX(4))
 BORE HOLE THROUGH CENTER
 1 LENGTH                                     .00253       .06157         R
A(1)=SIN(A(1))**2
 2 HEIGHT                                     .00796       1.99736      H
A(2)=ABS(COS(A(3)))+RNORM(3.29,2.4)
 3 WIDTH                                      .00534       .71998       H R
A(3)=SQRT(A(1))
 4 WEIGHT                                     .00318       5.3432
A(4)=ABS(COS(A(2)))+RNORM(5.4,.3)
 7     .043000     .000340
 8     .180000     3.65100
 6     .063000     1.09890
 9     .019500     .000027
 325
```

Figure 8A

Sample Input File

```
      SUBROUTINE PROCESS(I)
      COMMON/SIMCOM/A(5),NA(5),NX(10,5),KX(10,5),NPARTS(10),
     + XMEAN(10,5),XSTD(10,5),NATT(10,5),NCELLS(10,5),
     + CWIDTH(10,5),UPLIM(10,5),KCELL(10,5,11),NPROC,
     + DESCPROC,DESCATT,HIST,XX(10),REJ,TOL1,TOL2,
     + DMEAN(10,5),DSTD(10,5)
      CHARACTER DESCPROC(10)*66,DESCATT(10,5)*30,HIST(10,5)*1,
     + REJ(10,5)*1,TOL1(10,5)*10,TOL2(10,5)*10
      GO TO ( 1, 2, 3)I
1     CONTINUE
      A(1)=.03571
      XX(1)=SIN(A(1))
      A(2)=.17769
      A(3)=.657319
      A(4)=12.5786
      RETURN
2     CONTINUE
      A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(4.6,.09)))
      XX(4)=RNORM(3.495,.78)
      XX(3)=COS(A(3))
      IF(A( 1).LT..1234      )NX( 2,  1)=NX( 2,  1)+1
      IF(A( 1).GT..7984      )NX( 2,  1)=NX( 2,  1)+1
      A(2)=4*ABS(XX(3))+SQRT(XX(4))
      IF(A( 2).LT.1.324      )NX( 2,  2)=NX( 2,  2)+1
      IF(A( 2).GT.4.9734     )NX( 2,  2)=NX( 2,  2)+1
      RETURN
3     CONTINUE
      A(1)=SIN(A(1))**2
      IF(A( 1).LT..00253     )NX( 3,  1)=NX( 3,  1)+1
      IF(A( 1).GT..06157     )NX( 3,  1)=NX( 3,  1)+1
      A(2)=ABS(COS(A(3)))+RNORM(3.29,2.4)
      IF(A( 2).LT..00796     )NX( 3,  2)=NX( 3,  2)+1
      IF(A( 2).GT.1.99736    )NX( 3,  2)=NX( 3,  2)+1
      A(3)=SQRT(A(1))
      IF(A( 3).LT..00534     )NX( 3,  3)=NX( 3,  3)+1
      IF(A( 3).GT..71998     )NX( 3,  3)=NX( 3,  3)+1
      A(4)=ABS(COS(A(2)))+RNORM(5.4,.3)
      IF(A( 4).LT..00318     )NX( 3,  4)=NX( 3,  4)+1
      IF(A( 4).GT.5.3432     )NX( 3,  4)=NX( 3,  4)+1
      RETURN
      END
```

Figure 8B

Sample of Subroutine PROCESS

```
 3 4 2 4
INITIALIZE ATTRIBUTES
 1 LENGTH
 2 HEIGHT
 3 WIDTH
 4 WEIGHT
SHAPE THE PART
 1 LENGTH                                   .1234      .7984      H
 2 HEIGHT                                   1.324      4.9734     H R
BORE HOLE THROUGH CENTER
 1 LENGTH                                   .00253     .06157       R
 2 HEIGHT                                   .00796     1.99736    H
 3 WIDTH                                    .00534     .71998     H R
 4 WEIGHT                                   .00318     5.3432
 7     .043000     .000340
 8     .180000    3.651000
 6     .063000    1.098900
 9     .019500     .000027
 325
```

Figure 8C

Sample Scratch File

```
3 4 2 4
325INITIALIZE ATTRIBUTES
1 LENGTH
  .035710     .000001     .000000     .000000     0
2 HEIGHT
  .177689     .000019     .000000     .000000     0
3 WIDTH
  .657318     .000046     .000000     .000000     0
4 WEIGHT
12.578610     .000237     .000000     .000000     0
325SHAPE THE PART
                                      .1234       .7984      H
1 LENGTH
  .552311     .013986     .516600     .013986     89
7     .043000     .000340
 0     0     5    15    15    12     9   269
                                      1.324       4.9734     H R
2 HEIGHT
 3.828693     .013893    3.651003     .013892     0
8     .180000    3.651000
 83    58    83   101     0     0     0     0     0
325BORE HOLE THROUGH CENTER
                                      .00253      .06157       R
1 LENGTH
  .305453     .011194   -.246857      .003299     270
                                      .00796      1.99736    H
2 HEIGHT
 1.294890     .016017  -2.533801      .020741     0
6     .063000    1.098900
103    22    17    15    16    30   122
                                      .00534      .71998     H R
3 WIDTH
  .508956     .011988   -.148363      .011988     70
9     .019500     .000027
 0     0     0     0     1     6    10     7     7   294
                                      .00318      5.3432
4 WEIGHT
  .811413     .020086 -11.767190      .020082     0
```

Figure 8D

Sample Output File

****

**** SIMULATION SUMMARY REPORT ****

****

PROCESS 1 SHAPE THE PART

ATTRIBUTES MODIFIED BY PROCESS 1

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	89
2	HEIGHT	325	0

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.552311	.013986	.035710	.000001	.516600	.013986
2	3.828693	.013893	.177689	.000019	3.651003	.013892

PROCESS 2 BORE HOLE THROUGH CENTER

ATTRIBUTES MODIFIED BY PROCESS 2

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	270
2	HEIGHT	325	0
3	WIDTH	325	70
4	WEIGHT	325	0

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.305453	.011194	.552311	.013986	-.246857	.003299
2	1.294890	.016017	3.828693	.013893	-2.533801	.020741
3	.508956	.011988	.657318	.000046	-.148363	.011988
4	.811413	.020086	12.578610	.000237	-11.767190	.020082

Figure 8E - Entire Summary Report

```
******************************
** SIMULATION SUMMARY REPORT **
******************************
```

PROCESS 1 SHAPE THE PART

ATTRIBUTES MODIFIED BY PROCESS 1

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	89

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.552311	.013986	.035710	.000001	.516600	.013986

PROCESS 2 BORE HOLE THROUGH CENTER

ATTRIBUTES MODIFIED BY PROCESS 2

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	270

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.305453	.011194	.552311	.013986	-.246857	.003299

Figure 8F - Summary Report of a Particular Attribute

```
*****************************
** SIMULATION SUMMARY REPORT **
*****************************
```

PROCESS 2 BORE HOLE THROUGH CENTER

ATTRIBUTES MODIFIED BY PROCESS 2

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	270
2	HEIGHT	325	0
3	WIDTH	325	70
4	WEIGHT	325	0

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.305453	.011194	.552311	.013986	-.246857	.003299
2	1.294890	.016017	3.828693	.013893	-2.533801	.020741
3	.508956	.011988	.657318	.000046	-.148363	.011988
4	.811413	.020086	12.578610	.000237	-11.767190	.020082

Figure 8G - Summary Report of a Particular Process

```
****************************
** SIMULATION SUMMARY REPORT **
****************************
```

PROCESS 2 BORE HOLE THROUGH CENTER

ATTRIBUTES MODIFIED BY PROCESS 2

ATTRIBUTE #	DESCRIPTION	NO. PARTS MODIFIED	NO. PARTS OUT OF TOLERANCE
1	LENGTH	325	270

ATTRIBUTE #	CURRENT MEAN	CURRENT STD. DEV.	PREVIOUS MEAN	PREVIOUS STD. DEV.	DIFFERENCES OF MEANS	DIFFERENCES OF STD. DEVS.
1	.305453	.011194	.552311	.013986	-.246857	.003299

FIgure 8H - Summary Report of a Particular Porcess with a Particular Attribute

HISTOGRAM DATA FOR
PROCESS 1 ATTRIBUTE 1: LENGTH

OBSERVED NO. PARTS	RELATIVE NO. PARTS	CUMULATIVE NO. PARTS	UPPER CELL LIMIT
0	.000000	.000000	.000340
0	.000000	.000000	.043340
5	.015385	.015385	.086340
15	.046154	.061538	.129340
15	.046154	.107692	.172340
12	.036923	.144615	.215340
9	.027692	.172308	.258340
269	.827692	1.000000	INFINITY

325			

Figure 8I

Sample Histogram Table

HISTOGRAM DATA FOR
PROCESS 1 ATTRIBUTE 2: HEIGHT

OBSERVED NO. PARTS	RELATIVE NO. PARTS	CUMULATIVE NO. PARTS	UPPER CELL LIMIT
83	.255385	.255385	3.651000
58	.178462	.433846	3.831000
83	.255385	.689231	4.011000
101	.310769	1.000000	4.191000
0	.000000	1.000000	4.371000
0	.000000	1.000000	4.551000
0	.000000	1.000000	4.730999
0	.000000	1.000000	4.910999
0	.000000	1.000000	INFINITY

325			

Figure 8J

Sample Histogram Table

HISTOGRAM DATA FOR
PROCESS 2 ATTRIBUTE 2: HEIGHT

OBSERVED NO. PARTS	RELATIVE NO. PARTS	CUMULATIVE NO. PARTS	UPPER CELL LIMIT
103	.316923	.316923	1.098900
22	.067692	.384615	1.161900
17	.052308	.436923	1.224900
15	.046154	.483077	1.287900
16	.049231	.532308	1.350900
30	.092308	.624615	1.413900
122	.375385	1.000000	INFINITY
325			

Figure 8K

Sample Histogram Table

HISTOGRAM DATA FOR
PROCESS 2 ATTRIBUTE 3: WIDTH

OBSERVED NO. PARTS	RELATIVE NO. PARTS	CUMULATIVE NO. PARTS	UPPER CELL LIMIT
0	.000000	.000000	.000027
0	.000000	.000000	.019527
0	.000000	.000000	.039027
0	.000000	.000000	.058527
1	.003077	.003077	.078027
6	.018462	.021538	.097527
10	.030769	.052308	.117027
7	.021538	.073846	.136527
7	.021538	.095385	.156027
294	.904615	1.000000	INFINITY

325			

Figure 8L

Sample Histogram Table

Appendix II

Appendix II contains a detailed user's manual which will teach the user how to create a simulation of a sequential manufacturing operation and obtain summary reports and histogram tables.

User's Manual

This manual's primary purpose is to teach you how to create your own simulation of a sequential manufacturing operation and obtain the desired output reports. This will be accomplished by following the step by step detailed example that is provided. Examples, of all the files created and the types of reports that may be generated, are presented in Appendix I.

** WARNING **

It is IMPERATIVE that you READ THE ENTIRE MANUAL before attempting to run the programs. If an error is made while a program is executing, files you are using may be DAMAGED.

Getting Started

The first thing that you must do is boot the system and enter the current date and time. Entering the date and time is important because it will enable you to determine when a particular file was created or last modified. Once this is done the cursor should be in front of the prompt:

```
C:\WORK>
```

If you type

C:\WORK>RUNINPUT

After a few seconds you should see:

SIMULATION OF A

PRODUCTION LINE

VERSION 1.0

BY DR. L.J. PLEBANI AND JILL B. WOOTTEN

This will remain on the screen for only a couple of seconds.

The next message will be:

REMINDER ON SPECIAL DETAILS

1. ESC (escape)- this key can be entered whenever a choice is given. It is used to "escape" when in trouble by getting instructions, starting over or quiting the program.

2. If a list is longer than can be printed on the screen, only a portion will be printed at a time. By hitting the <CR> (carriage return) the list continues.

3. Ctrl-C can be entered when the user is unable to use ESC and needs to exit the program. If these keys are used all information entered during the session will be lost. The only time a ctrl-C MAY NOT be used is when a file name is being entered.

Nothing else will happen until you hit the carriage return.

Determining the Input File

Once you have entered a carriage return, you will be presented with the options contained in Menu A:

```
1. Load an OLD Input File
2. Create a NEW Input File
3. List all old Input Files

Enter (1,2 or 3): \
```

If you press the "ESC" key (which appears as a backward slash on the display), the following will appear on the screen:

```
Type S to Start Over
Type H to get Help
Type Q to Quit the Program
Type C to Continue

Enter (S,H,Q or C): C
```

When an H is entered, a list of instructions will be displayed then Menu A will appear again.
When an S is entered, the program will start over from the very beginning.

When a Q is entered, the program will end and you will see the prompt:

```
C:\WORK>
```

When a C is entered, the program will continue as if "ESC" was never entered and Menu A will be displayed.

If a 3 is entered from Menu A, a list of all the old input files will be displayed. If it is the first time the program is being executed only the file name EXAMPLE will appear. Each time a new file is created it is added to this list.

Creating a New File

To create a new file enter the number 2 from Menu A.

```
1. Load an OLD Input File
2. Create a NEW Input File
3. List all old Input Files

Enter (1,2 or 3): 2
```

The following will appear on the screen:

```
Enter the Name of the File according to the
following guidelines:

1. The name can be up to 8 alphanumeric characters

2. The first character must be a letter

3. No special characters may be used (*,/,?,! etc.)

Enter the File Name : EXAMPLE
```

For this sample session the file name EXAMPLE is used. After a name is entered a check is made to determine if the name already exists. If it does the following message appears:

```
EXAMPLE  ALREADY EXISTS

USE IT ANYWAY? (Y/N) : Y
```

If you enter 'N' Menu A will return.

If you enter 'Y', a warning is displayed.

```
IF EXAMPLE  IS USED THE PRESENT FILE WILL BE DESTROYED

USE IT ANYWAY? (Y/N) :
```

If you answer 'N', Menu A is displayed. If you answer 'Y' all the contents of that file are cleared and a new file is created.

If the file name you entered does not exist a check is made to insure that the name entered does not begin with a number and does not contain special characters.
When a name contains special characters, the following message appears.

```
THE FILE NAME CAN NOT CONTAIN SPECIAL CHARACTERS
```

When a name has a number as its first character, the following message appears.

```
THE FILE NAME CAN NOT BEGIN WITH A NUMBER
```

In either case you must hit the carriage return in order to continue. You will then be asked to enter another name.

Enter the Part Attributes

Once the new file name has been determined, the following is displayed:

```
Enter the Description for Each Attribute
of the Part being Manufactured

Attribute #  Description
     1       LENGTH
     2       HIEGHT
     3       WIDTH
     4       DONE
```

You are given the attribute # as a prompt and are expected to enter the description of each one. When you are finished entering all the part attributes, type "DONE" for the description.

You must then enter the initial values for these attributes. The screen appears as follows:

```
Enter the Initial Value (Equation A(#)=value)
of the Part being Manufactured

Attribute #  Description                Initial Value
     1       LENGTH                     A(1)=.03571
     2       HIEGHT                     A(2)=.17769
     3       WIDTH                      A(3)=.657319
```

Once all of the Initial Values have been entered, all the information for the part attributes is displayed and you are asked if everything is correct.

Attribute #	Description	Initial Value
1	LENGTH	A(1)=.03571
2	HIEGHT	A(2)=.17769
3	WIDTH	A(3)=.657319

IS EVERYTHING CORRECT? (Y/N): N

If you answer 'N' the following dialogue occurs:

MODIFY ATTRIBUTE #: 2

ENTER THE DESCRIPTION : HEIGHT

ENTER THE INITIAL VALUE : A(2)=.17769

MODIFY ANOTHER ATTRIBUTE? (Y/N): N

If you type 'Y' and wish to modify another attribute, then the same dialogue will occur again. If you type 'N' then the attributes appear on the screen with all the changes made.

Attribute #	Description	Initial Value
1	LENGTH	A(1)=.03571
2	HEIGHT	A(2)=.17769
3	WIDTH	A(3)=.657319

IS EVERYTHING CORRECT? (Y/N): Y

ADD ANOTHER ATTRIBUTE? (Y/N): Y

If you answer 'Y' and wish to add another attribute you are given the next attribute number as a prompt and the following appears:

```
ENTER INFORMATION FOR ATTRIBUTE #: 4
ENTER THE DESCRIPTION : WEIGHT
ENTER THE INITIAL VALUE : A(4)=12.5786
```

Once all the attributes have been added, they are listed and you are given the chance to modify them again.

Attribute #	Description	Initial Value
1	LENGTH	A(1)=.03571
2	HEIGHT	A(2)=.17769
3	WIDTH	A(3)=.657319
4	WEIGHT	A(4)=12.5786

```
IS EVERYTHING CORRECT? (Y/N): Y
ADD ANOTHER ATTRIBUTE? (Y/N): N
```

If everything is correct and you do not wish to enter another attribute you will be given the option of entering additional initial equations. These equations are used to further describe the system and can be any valid FORTRAN statement. The variable XX(I) is defined as a global variable that may be used in these equations.

If you type 'Y' to enter additional initial equations the following screen dialogue occurs:

```
Do you want to enter Additional Initial Equations? (Y/N): Y

ENTER EQUATION  1
XX(1)+SIN(A(1))

XX(1)+SIN(A(1))
IS THIS CORRECT? (Y/N): N
```

If it is not correct type 'N', and re-enter the equation.

```
ENTER THE CORRECTED EQUATION
XX(1)=SIN(A(1))

XX(1)=SIN(A(1))
IS THIS CORRECT? (Y/N): Y

ENTER ANOTHER EQUATION? (Y/N): N
```

If it is correct type 'Y' and enter all other initial equations.

Enter the Processes

You must now enter the processes. You are given the process number as a prompt and asked to enter a description of each. The screen appears as follows:

```
ENTER THE DESCRIPTION OF EACH PROCESS

PROCESS #   DESCRIPTION
    1       MOLD THE PART
    2       BORE HOLE THROUGH CENTER
    3       DONE
```

Once you have finished entering all the process descriptions, type the word "DONE" as the next decription. All of the process numbers and descriptions will then be displayed to insure that they are correct.

```
PROCESS #   DESCRIPTION
    1       MOLD THE PART
    2       BORE HOLE THROUGH CENTER

 IS EVERYTHING CORRECT? (Y/N): N
```

If an 'N' is entered the following dialogue occurs:

```
ENTER THE # OF THE PROCESS TO BE MODIFIED: 1

ENTER THE DESCRIPTION: SHAPE THE PART
```

The processes and their descriptions are displayed again and you have the option of correcting another error.

```
PROCESS #   DESCRIPTION
    1       SHAPE THE PART
    2       BORE HOLE THROUGH CENTER
```

```
IS EVERYTHING CORRECT? (Y/N): Y
```

Enter 'Y' once all errors have been eliminated. You will then see the following question:

```
ADD ANOTHER PROCESS? (Y/N):
```

If you wish to add another process type 'Y'. The following prompt will appear with the next process number.

```
ENTER THE DESCRIPTION FOR PROCESS  3

HEAT THE PART
```

Once the description is entered, all the processes are displayed so you may check for errors.

```
PROCESS #   DESCRIPTION
    1       SHAPE THE PART
    2       BORE HOLE THROUGH CENTER
    3       HEAT THE PART
```

```
IS EVERYTHING CORRECT? (Y/N): Y

ADD ANOTHER PROCESS? (Y/N):N
```

After all processes have been entered and all errors have been corrected, enter 'Y' and 'N', respectively, to the above questions.

It is now time to determine which attributes are modified by each process. The following is an example of the dialogue which typically occurs.

```
DETERMINE WHICH ATTRIBUTES ARE MODIFIED BY
PROCESS  1 SHAPE THE PART


DOES PROCESS  1 MODIFY                        ? (Y/N): Y
ATTRIBUTE 1 LENGTH
```

If you answer 'N', then the following will appear:

```
DOES PROCESS  1 MODIFY                        ? (Y/N):
ATTRIBUTE 2 HEIGHT
```

Each time an 'N' is entered, you will be asked if the next attribute is modified by that process.

If you enter 'Y', the following information is entered. Be sure to include a decimal point when entering the tolerances, cell width and upper limit on the first cell. The number of cells, however must be an integer.

```
ENTER THE MODIFYING EQUATION

A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(2.1,.02))

ENTER THE LOWER TOLERANCE .1234

ENTER THE UPPER TOLERANCE .7984
```

```
REJECT PARTS OUT OF TOLENANCE? (Y/N): N

COLLECT HISTOGRAM DATA? (Y/N): Y

ENTER THE # OF CELLS: 7

ENTER THE CELL WIDTH: .27597

ENTER THE UPPER LIMIT OF THE FIRST CELL: .00034

DO YOU WANT TO ENTER ADDITIONAL EQUATIONS? (Y/N): Y

ENTER THE EQUATION

XX(3)=COS(A(2))


IS THIS CORRECT? (Y/N): N

ENTER THE CORRECTED EQUATION

XX(4)=RNORM(3.45,.98)

XX(4)=RNORM(3.45,.98)

IS THIS CORRECT? (Y/N): Y

ENTER ANOTHER EQUATION? (Y/N): N
```

This information is entered for each attribute that is modified by a particular process. Once all information, for all the attributes modified by this process, is entered it is listed on the screen as follows:

```
ATTRIBUTES FOR PROCESS   1

                                                          HIST  REJECT
ATTRIBUTE  DESCRIPTION          TOL1          TOL2         H
    1      LENGTH               .1234         .7984        H       R
    2      HEIGHT               1.324         4.9734

MODIFYING EQUATION
A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(2.1,.02))
A(2)=4*ABS(XX(3))+SQRT(XX(4))


IS EVERYTHING CORRECT? (Y/N): N
```

If you answer 'N' then the following appears:

```
MODIFY ATTRIBUTE # : 1
```

You are then asked to enter all the attribute information as before. Once this is done, the data is displayed on the screen.

```
ATTRIBUTES FOR PROCESS   1

                                                          HIST  REJECT
ATTRIBUTE  DESCRIPTION          TOL1          TOL2         H
    1      LENGTH               .1234         .7984        H       R
    2      HEIGHT               1.324         4.9734

MODIFYING EQUATION
A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(4.6,.09))
A(2)=4*ABS(XX(3))+SQRT(XX(4))


IS EVERYTHING CORRECT? (Y/N): Y
```

If you answer 'Y' and histogram specifications were entered for one or more of the attributes, the following appears on the screen:

```
HISTOGRAM SPECIFICATIONS FOR THE ATTRIBUTES OF
PROCESS  1

ATTRIBUTE DESCRIPTION     # CELLS   CELL WIDTH UPPER LIMIT
    1      LENGTH            7        .454300     .003400
    2      HEIGHT            8       1.313810     .036250

IS EVERYTHING CORRECT? (Y/N): N
```

If 'N' is entered , you will see the following:

```
MODIFY ATTRIBUTE # : 1
```

You are given the option of entering histogram data for that attribute. If you enter 'Y', data is entered as before. All attributes are listed again and you are given the option of making corrections. Once all corrections have been made, you will be asked:

```
DO YOU WANT TO ADD ANOTHER ATTRIBUTE? (Y/N): N
```

If you answer 'Y', you will be asked to enter all the information for the attribute you selected. If you answer 'N', you will be asked to determine which attributes are modified by the remaining processes.

Once it has been determined which attributes are modified by all processes the following will appear:

```
ENTER THE NUMBER OF PARTS TO BE PROCESSED: 300

300 PARTS
IS THIS CORRECT? (Y/N): N
```

If you enter 'N', you will again be asked to enter the number of parts.

Once you answer 'Y', the information you have entered will be saved under the file name you specified in the beginning. The red light on the hard disk drive will blink for a couple of minutes while a FORTRAN subroutine is constructed from the information you entered. The subroutine is then compiled and linked to the simulation program. A sample copy of this subroutine can be seen in Appendix I.

<u>Modify an Existing File</u>

To modify an existing file enter the number 1, from Menu A:

```
1. Load an OLD Input File
2. Create a NEW Input File
3. List all old Input Files

Enter (1,2 or 3): 1
```

After a 1 is entered the following will appear:

```
Enter the Name of the File to be used: EXMPLE
```

If the file name entered does not exist, the following error message will be displayed:

```
EXMPLE   DOES NOT EXIST AS AN OLD FILE
```

After you hit return, Menu A will appear again. Once you have entered a file name correctly, you are given the option to modify the file.

```
MODIFY THE FILE? (Y/N): Y
```

If the simulation is to be run with the file exactly the way it is, answer 'N' to this question and proceed with the simulation. If the file must be modified enter 'Y' and Menu B will appear:

```
1. MODIFY A PROCESS
2. ADD A PROCESS
3. DELETE A PROCESS
4. CHANGE THE # OF PARTS TO BE PROCESSED
5. FINISHED MODIFYING

ENTER (1,2,3,4 OR 5): 4
```

If you enter 4, the following will be displayed:

```
THE # OF PARTS CURRENTLY LISTED IS  325
ENTER THE CORRECT # OF PARTS: 350
```

Once you enter the corrected number of parts, the number will be displayed as follows:

```
PROCESS  350 PARTS
IS THIS CORRECT? (Y/N): Y
```

If you enter 'N', you will be asked to enter the corrected number again. If you enter 'Y', Menu B will reappear.

```
1. MODIFY A PROCESS
2. ADD A PROCESS
3. DELETE A PROCESS
4. CHANGE THE # OF PARTS TO BE PROCESSED
5. FINISHED MODIFYING

ENTER (1,2,3,4 OR 5): 3
```

If you enter 3, the processes and descriptions will be listed as follows:

```
PROCESS DESCRIPTION
   0    INITIALIZE ATTRIBUTES
   1    SHAPE THE PART
   2    BORE HOLE THROUGH CENTER
   3    HEAT THE PART
ENTER THE # OF THE PROCESS TO BE DELETED: 2
```

Once you enter the number of the process to be deleted, a check is made to be sure the correct number was entered.

```
PROCESS  2 IS TO BE DELETED? (Y/N): N
```

If you answer 'N', you are given the following option:

```
DELETE A DIFFERENT PROCESS? (Y/N): Y
```

If you answer 'N', Menu B is displayed. If you answer 'Y', the processes and descriptions are displayed again.

```
PROCESS DESCRIPTION
   0     INITIALIZE ATTRIBUTES
   1     SHAPE THE PART
   2     BORE HOLE THROUGH CENTER
   3     HEAT THE PART

ENTER THE # OF THE PROCESS TO BE DELETED: 3

PROCESS  3 IS TO BE DELETED? (Y/N): Y
```

If you answer 'Y', the process is deleted and you are asked the following:

```
DELETE ANOTHER PROCESS? (Y/N): N
```

If you answer 'Y', the whole cycle begins again. If you answer 'N', Menu B reappears.

```
1. MODIFY A PROCESS

2. ADD A PROCESS

3. DELETE A PROCESS

4. CHANGE THE # OF PARTS TO BE PROCESSED

5. FINISHED MODIFYING

ENTER (1,2,3,4 OR 5): 2
```

If you enter 2, the processes and descriptions are listed as follows:

```
PROCESS DESCRIPTION
   0     INITIALIZE ATTRIBUTES
   1     SHAPE THE PART
   2     BORE HOLE THROUGH CENTER

ENTER THE NUMBER OF THE PROCESS WHICH THE NEW
PROCESS IS TO FOLLOW: 1
```

Once you enter the appropriate number above, you are asked for the following:

```
ENTER THE DESCRIPTION FOR THE NEW PROCESS

HEAT THE PART
```

After the description is entered you must type in all the information about the attibutes modified by the new process. This is done exactly the same way as when the file was initially created (see Enter the Processes). Once the information is added you are given the following option:

```
ADD ANOTHER PROCESS? (Y/N): N
```

If you answer 'Y', the whole cycle is repeated. If you enter 'N', Menu B is displayed.

```
1. MODIFY A PROCESS

2. ADD A PROCESS

3. DELETE A PROCESS

4. CHANGE THE # OF PARTS TO BE PROCESSED

5. FINISHED MODIFYING


ENTER (1,2,3,4 OR 5): 1
```

If you choose 1, the processes and their descriptions are listed as follows:

```
PROCESS #  DESCRIPTION
    0      INITIALIZE ATTRIBUTES
    1      SHAPE THE PART
    2      HEAT THE PART
    3      BORE HOLE THROUGH CENTER

ENTER THE # OF THE PROCESS TO BE MODIFIED: 1
```

Once the process number is entered the following options appear:

```
1. MODIFY THE PROCESS DESCRIPTION

2. MODIFY THE ATTRIBUTES


ENTER (1 OR 2): 1
```

If you enter 1, you are given the following prompt:

```
ENTER THE CORRECT DESCRIPTION FOR PROCESS  1
MOLD THE PART
```

Once the description is entered, it is displayed again.

```
MOLD THE PART
IS THIS CORRECT? (Y/N): Y
```

If you answer 'Y', Menu B appears. If you answer 'N', you will be given a prompt to enter the corrected description. Given the following options:

```
1. MODIFY THE PROCESS DESCRIPTION
2. MODIFY THE ATTRIBUTES

ENTER (1 OR 2): 2
```

If you enter 2, Menu C will appear on the screen:

```
1. ADD AN ATTRIBUTE
2. MODIFY AN ATTRIBUTE
3. DELETE AN ATTRIBUTE
4. MODIFY ADDITIONAL EQUATIONS

ENTER (1,2,3 OR 4):
```

If you enter 1, the program determines which attibutes have not been modified by the particular process specified and asks you which ones are to be modifed. Information is entered for the attributes as before. Once you are done adding attributes Menu B is displayed.

If you enter 2 from Menu C, a list of all the attributes is displayed as follows:

```
ATTRIBUTES FOR PROCESS  1

                                                        HIST REJECT
ATTRIBUTE DESCRIPTION       TOL1         TOL2            H
    1      LENGTH           .1234        .7984           H      R
    2      HEIGHT           1.324        4.9734

MODIFYING EQUATION
A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(4.6,.09))
A(2)=4*ABS(XX(3))+SQRT(XX(4))

IS EVERYTHING CORRECT? (Y/N): N
```

You are now able to make all the necessary modifications. The procedure is the same as when the file is first created. If you enter 3 from Menu C, a list of all the attributes is displayed.

```
ATTRIBUTES FOR PROCESS  1

                                                        HIST REJECT
ATTRIBUTE DESCRIPTION       TOL1         TOL2            H
    1      LENGTH           .1234        .7984           H      R
    2      HEIGHT           1.324        4.9734

MODIFYING EQUATION
A(1)=2*ABS(XX(1))+ABS(SIN(RNORM(4.6,.09))
A(2)=4*ABS(XX(3))+SQRT(XX(4))

ENTER THE ATTRIBUTE # TO BE DELETED: 2

DELETE ANOTHER ATTRIBUTE? (Y/N): N
```

Once you enter the attribute number, it is deleted and you are given the option to delete another attribute. Then all the remaining attributes are listed and you are given the option of modifying them.

If you enter 4 from Menu C, the additional equations are listed.

```
ADDITIONAL EQUATIONS UNDER PROCESS  1 ATTRIBUTE 1

EQUATION #
    1        XX(3)=COS(A(2))
    2        XX(4)=RNORM(3.45,.98)
```

You are then given the option to delete an equation.

```
DELETE AN EQUATION? (Y/N): Y
```

If you enter 'Y', the following appears:

```
ENTER THE # OF THE EQUATION TO BE DELETED: 1
```

Once you enter the equation number, it is deleted and the remaining equations are listed.

If you enter 'N' and do not wish to delete an equation, the next option appears:

```
ADD AN EQUATION? (Y/N): Y
```

If you enter 'Y', the following prompt is given:

```
ENTER THE EQUATION
XX(3)=COS(A(3))
```

Once the equation is entered, all additional equations are listed.

```
ADDITIONAL EQUATIONS UNDER PROCESS  1 ATTRIBUTE 1

EQUATION #
     1       XX(4)=RNORM(3.45,.98)
     2       XX(3)=COS(A(3))

DELETE AN EQUATION? (Y/N): N
ADD AN EQUATION? (Y/N): N
IS EVERYTHING CORRECT? (Y/N):
```

If you enter 'N' for all the above questions, the following appears:

```
ENTER EQUATION # TO BE MODIFIED: 1
ENTER THE CORRECTED EQUATION
XX(4)=RNORM(3.495,.78)
```

Once the corrected equation is entered, all the equations are listed once again.

```
ADDITIONAL EQUATIONS UNDER PROCESS  1 ATTRIBUTE 1

EQUATION #
    1      XX(4)=RNORM(3.495,.78)
    2      XX(3)=COS(A(3))

DELETE AN EQUATION? (Y/N): N

ADD AN EQUATION? (Y/N): N

IS EVERYTHING CORRECT? (Y/N): Y
```

If you answer 'Y' and another attribute for the specified process has additional equations, they are listed and the cycle repeats.

If you answer 'Y' and another attribute for the specified process does not have additional equations you may enter additional equations for that attribute.

Once you have finished adding, deleting or modifying the additional equations for this particular attribute, Menu B is displayed.

```
1. MODIFY A PROCESS

2. ADD A PROCESS

3. DELETE A PROCESS

4. CHANGE THE # OF PARTS TO BE PROCESSED

5. FINISHED MODIFYING


ENTER (1,2,3,4 OR 5): 5
```

When you are finished editing the file, enter 5. The file will be saved and a FORTRAN subroutine will be created, compiled and then linked with the simulation program. A sample copy of this subroutine can be seen in Appendix I.

Running the Simulation

Once the FORTRAN subroutine has been constructed, compiled and linked to the simulation program and you will see the prompt:

```
C:\WORK>
```

You are now ready to run the simulation. In order to do this just type the following at the prompt:

```
C:\WORK>SIMULATE
```

Once you have entered the above, you will be presented with Menu D:

```
1. Use an OLD Output File as a NEW Output File
2. Create a NEW Output File Name
3. List all old Output File Names

Enter (1,2 or 3): 1
```

Remember that the output file name CAN NOT have the same name as the input file.

If you enter 3, a list of all output files will be displayed and then the above options will appear again.

If you enter 1, the following will appear:

```
Enter the Name of the File to be used: EXAMP
```

If the name does not exist, the following message will be displayed.

```
EXAMP    DOES NOT EXIST AS AN OLD FILE
```

You must enter a carriage return in order to continue. Once you do so Menu D will reappear. Remember that if you use an old file name, the contents of that file will be destroyed and replaced with new information.

If you enter 2, the following guidelines will be displayed:

```
Enter the Name of the File according to the
following guidelines:

1. The name can be up to 8 alphanumeric characters

2. The first character must be a letter

3. No special characters may be used (*,/,?,! etc.)

Enter the File Name : OUTPUT
```

If the name entered already exists the following message will appear:

```
OUTPUT    ALREADY EXISTS
USE IT ANYWAY? (Y/N) : Y
```

If you answer 'N', Menu D will appear again. If you answer 'Y', the next warning appears:

```
IF OUTPUT    IS USED THE PRESENT FILE WILL BE DESTROYED
USE IT ANYWAY? (Y/N) : N
```

If you answer 'N', Menu D will reappear. If you answer 'Y', the simulation will continue.

When you enter a name that does not exist, the new name is checked to be sure that it adheres to the guidelines given. If it does not, one of the following messages will appear:

```
THE FILE NAME CAN NOT BEGIN WITH A NUMBER

THE FILE NAME CAN NOT CONTAIN SPECIAL CHARACTERS
```

In either case a carriage return is needed to continue and you must enter another name.

Once a name has been entered correctly, the simulation is ready to begin. The following message appears:

SIMULATION

IS

RUNNING

This remains on the screen a few seconds. The following information is then displayed while the simulation runs.

OF PARTS ALREADY PROCESSED:

OF PARTS REJECTED:

When the simulation is finished the following prompt will appear at the top of the screen.

C:\WORK>

Summary Reports

In order to see the results of the simulation type the following at the prompt:

```
C:\WORK>SUMMARY
```

The following options will be displayed:

```
1. LOAD AN OUTPUT FILE
2. LIST ALL OUTPUT FILES

ENTER (1 OR 2): 1
```

If you enter 2, a list of all output files will be displayed.

When you enter 1, you will be asked to enter a file name.

```
Enter the Name of the File to be used: OTPUT
```

If the name does not exist, the following message will be displayed:

```
OTPUT    DOES NOT EXIST AS AN OLD FILE
```

When you enter a carriage return to continue, the options will be displayed again.

Once a file name has been entered correctly, the following title appears for a few seconds.

SIMULATION

SUMMARY

REPORTS

Then the following message is displayed:

```
SPECIAL NOTE:

IF THE USER WANTS TO END THE PROGRAM WHILE
IN THE MIDDLE TYPE CTRL-C.  NO DATA WILL BE LOST.
TO GET BACK INTO THE PROGRAM TYPE SUMMARY.
```

You must type a carriage return to continue, then Menu E will appear:

```
1. Display Summary Reports

2. Display Histogram Data

3. End the Program


Enter (1,2 or 3): 1
```

If a 3 is entered the program is ended.

If a 1 is entered from Menu E, Menu F will appear:

1. Display the Entire Summary Report
2. Display Selected Parts of the Summary Report

Enter (1 or 2): 1

If you choose 1, the following is displayed:

1. Display Summary Report
2. Print Summary Report
3. Display and Print Summary Report

Enter (1,2 or 3):

Once a choice is made the entire report will be routed to the correct device and then Menu E will reappear. If 1 is entered from Menu E, Menu F is displayed.

1. Display the Entire Summary Report
2. Display Selected Parts of the Summary Report

Enter (1 or 2): 2

If you choose 2 from Menu F, the following options appear:

1. Display Summary Report for a particular Attribute
2. Display Summary Report for a particular Process
3. Display Summary Report for a particular Process with a particular Attribute

Enter (1,2 or 3):

When you choose one of these options you are asked for the appropriate information and if the report should be printed. You are asked if you wish to see another report and if you answer 'N' Menu E will reappear.

1. Display Summary Reports
2. Display Histogram Data
3. End the Program

Enter (1,2 or 3): 2

If you choose option 2 from Menu E, Menu G is displayed:

1. Display All Histogram Data
2. Display Selected Histogram Data

Enter (1 or 2): 1

If you choose 1 from Menu G, the following options appear:

1. Print and Display all Histogram Data
2. ONLY Display Histogram Data
3. ONLY Print Histogram Data
4. Display all and Print Selected Histogram Data

Enter (1,2,3 or 4):

Once a choice is made the histograms are routed to the correct device and Menu E will appear.

If you choose 2 from Menu G, display selected histogram data, the following options appear:

1. Display all Histogram Data for a particular Attribute
2. Display all Histogram Data for a particular Process
3. Display Selected Histograms

Enter (1,2 or 3):

When you select one of these options, you are asked for the appropriate information and if you want a printout. You are also asked if you wish to see more data and if you answer

'N' Menu E will be displayed.

An example of each of these different types of reports can be seen in Appendix I.

NOTES:

1) All file names must be entered in capital letters.

2) A ctrl-C must not be used to exit the program when entering a file name because the control character is placed in the list of files and causes an error the next time the program is executed.

3) The number of processes, the number of attributes and the number of equations that may be entered is restricted by the size to which the respective arrays are dimensioned.

4) There are five random number generators available, which may be used in the modifying and additional equations. These include:

a) Function RNORM(XMN,ST)- RNORM is set equal to a random number from a normal distribution with mean equal to XMN and standard deviation equal to ST. Both of these numbers must be real.

b) Function UNFRM(ULO,HI)- UNFRM is set equal to a random number from a uniform distribution with the low end of the interval equal to ULO and the upper end equal to HI. Both of these numbers must be real.

c) Function TRIAG(TLO,TMD,THI)- TRIAG is set equal to a random number from a triangular distribution with the low end of the interval equal to TLO, the upper end equal to THI and the mode equal to TMD. All three of these values must be real.

d) Function EXPON(XMN)- EXPON is set equal to a random number from an exponential distribution with the mean equal to XMN. This number must be real.

e) Function RAND(I)- RAND is set equal to a random number between zero and one from a uniform distribution. The argument I is a dummy variable, however it must be assigned an integer value whenever the function is used.

Appendix III - Biography

On November 11, 1961, Jill B. Wootten was born in Belleville, New Jersey to Helen and Raymond Wootten. In May of 1983, she graduated Cum Laude, from Drew University, Madison, New Jersey, with a BA in mathematics. She began her graduate studies at Lehigh University, Bethlehem, Pennsylvania, in August of 1985, with the Mathematics Department and later transfered to the Department of Industrial Engineering. She held a position as a teaching assistant in the Math Department for the academic year 1983-84 and in the Accounting Department for 1984-85. She will graduate with an MSIE, from the Department of Industrial Engineering, in June 1985.

A Simulation for the Evaluation of Tolerances in a Sequential Manufacturing Operation

by
Jill B. Wootten

Abstract

A user configurable simulation is presented which models sequential manufacturing operations with imbedded inspections stations. The machines are defined by sets of modifying equations which are used to alter specific attributes of a part as it is processed. When a part leaves a machine, an inspection may be performed to determine if the attibutes are within specified tolerances. If one of the attributes is out of tolerance, it is recorded as a reject and is removed from the line if so instructed. The number of rejections and statistics on an attribute are collected for each attribute modified by each machine. Data for a histogram for each attribute may also be collected. Output reports are generated from the statistics and histogram data that is compiled and can be constructed to fit the specific needs of the user. These reports provide the basis for critical decisions such as changing a distribution or tolerance range. The simulation can be easily modified to accommodate these decisions. The user, in effect, has the ability to create and modify his own simulation which is easy to implement and is informative as well.

ces

l which
imbedded
by sets of
.ific
ι part leaves
:ermine if
If one of
led as a
ıcted. The
bute are
achine. Data
collected.
s and
tructed to
rts provide

the basis for critical decisions such as changing a distribution or tolerance range. The simulation can be easily modified to accommodate these decisions. The user, in effect, has the ability to create and modify his own simulation which is easy to implement and is informative as well.

Appendix III - Biography

On November 11, 1961, Jill B. Wootten was born in Belleville, New Jersey to Helen and Raymond Wootten. In May of 1983, she graduated Cum Laude, from Drew University, Madison, New Jersey, with a BA in mathematics. She began her graduate studies at Lehigh University, Bethlehem, Pennsylvania, in August of 1985, with the Mathematics Department and later transfered to the Department of Industrial Engineering. She held a position as a teaching assistant in the Math Department for the academic year 1983-84 and in the Accounting Department for 1984-85. She will graduate with an MSIE, from the Department of Industrial Engineering, in June 1985.